

MSC

2.º  
CICLO

FCUP  
2013

U. PORTO

Sistema de Recomendação, Relatórios e Análise de  
Tendências de Rede Social Desportiva Playnify

Gabriel de Jesus

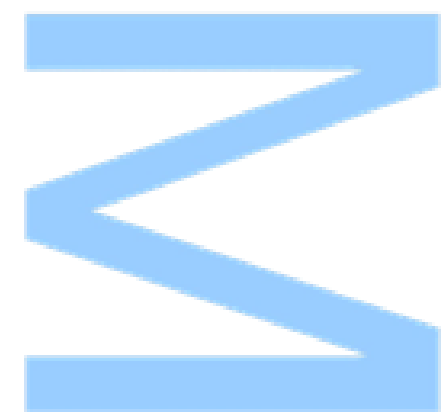
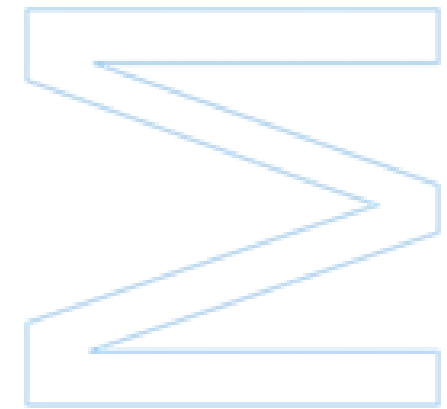
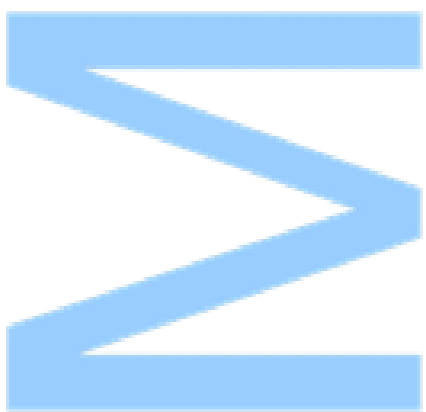
FC



# Sistema de Recomendação, Relatórios e Análise de Tendências de Rede Social Desportiva Playnify

Gabriel de Jesus

Dissertação de Mestrado apresentada à  
Faculdade de Ciências da Universidade do Porto em  
Ciência de Computadores  
2013







# Sistema de Recomendação, Relatórios e Análise de Tendências de Rede Social Desportiva Playnify

**Gabriel de Jesus**

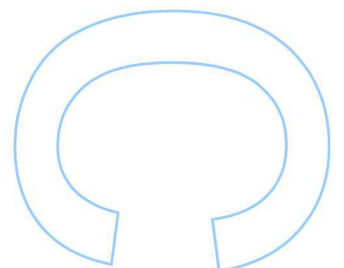
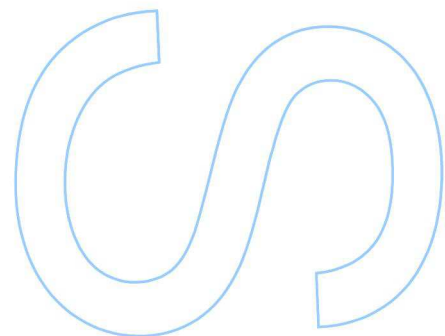
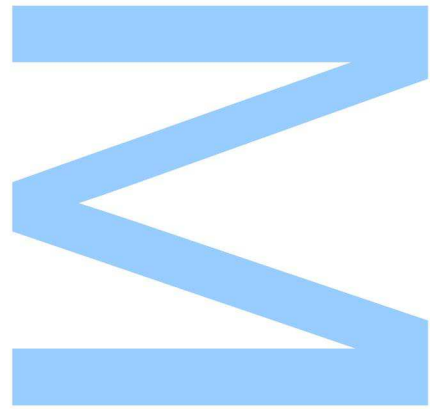
Data Mining e Processamento Avançado de Dados  
Ciência de Computadores  
2013

**Orientador**

Ricardo Jorge Veríssimo Ferrolho, Web/Multimedia Designer/Developer,  
Ubiwhere, Lda

**Coorientador**

Sabine Babette Broda, Professora Associada, Faculdade de Ciência da  
Universidade do Porto



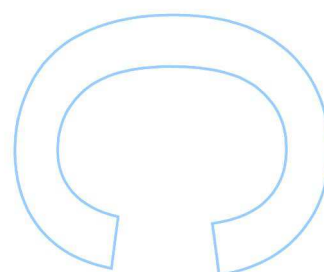
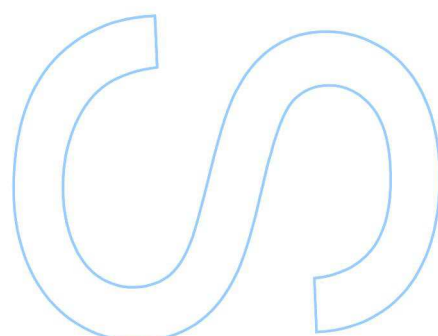
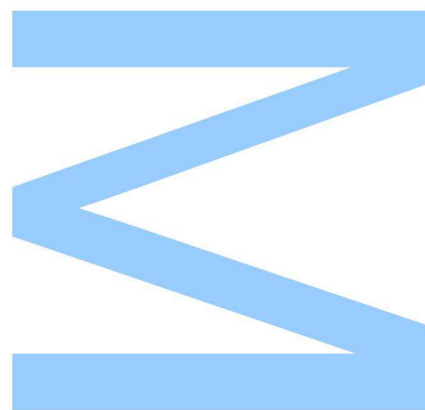




Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, \_\_\_\_/\_\_\_\_/\_\_\_\_





# Agradecimentos

Em primeiro lugar à minha família que sempre me apoio e ajudou em tudo o que puderam, sendo sempre um grande exemplo de trabalho e humildade.

À minha orientadora Professora Sabine Babette Broda por todo o tempo despendido comigo e pela preciosa ajuda e disponibilidade prestada na escrita deste relatório.

À Ubiwhere por me ter concedido a oportunidade de realizar este estágio. A todos os colaboradores da Ubiwhere pela forma como me receberam, especialmente ao Ricardo Ferrolho por toda a ajuda e disponibilidade prestada durante desenvolvimento do projeto.

Ao Professor Gabriel de Sousa Torcato David e Professor Alípio Mário Guedes Jorge pelas motivações e sugestões dadas durante da realização deste projeto.

Por último, mas não menos importante, a todos os meus amigos que sempre me apoiaram e motivaram a seguir em frente.

A todos o meu muito obrigado.





# Resumo

A empresa Ubiwhere lançou no final de 2012 um produto novo que é uma plataforma de rede social direcionada a desportistas amadores – o Playnify. Com o constante aumento do número de utilizadores da plataforma, tornou-se importante que os responsáveis/administradores da plataforma pudessem conhecer os perfis/atividades/preferências dos seus utilizadores, para com base neste conhecimento poder fornecer um serviço útil e agradável e assegurar a qualidade da utilização da plataforma.

Para responder a estas necessidades, desenvolvemos um Sistema de Recomendação e um Sistema de Relatórios e Análise de Tendências. O Sistema de Recomendação (versão Demo) é desenvolvido para substituir o atual sistema de recomendação aleatório do Playnify. Este sistema é dirigido aos utilizadores para recomendar-lhes outros utilizadores (desportistas) que eles provavelmente gostariam de adicionar à sua rede de desportistas, como também eventos e instalações desportivas adequados aos seus perfis desportivos. O Sistema de Relatórios e Análise de Tendências é dirigido aos responsáveis da plataforma para monitorizarem as atividades, os hábitos e as tendências dos utilizadores do Playnify.

Para desenvolver o Sistema de Recomendação utilizámos SQL na implementação dos algoritmos de recomendação, e ASP.NET para o desenvolvimento do protótipo. Relativamente ao Sistema de Relatórios e Análise de Tendências desenvolvemos primeiro os relatórios no Crystal Reports do SAP para Microsoft Visual Studio 2010. Em seguida integramos os relatórios com as páginas *web* do sistema, desenvolvidas em ASP.NET.

A versão Demo do Sistema de Recomendação será integrada mais tarde com o Playnify que se encontra em fase de desenvolvimento contínuo. O Sistema de Relatórios e Análise de Tendências já está no ambiente de produção.



# Abstract

The Ubiwhere company launched at the end of 2012 a new product which is a social networking platform targeted to amateur sportsmen – the Playnify. With the constant increasing of the number of platform users, it has become important that the administrators of the platform know the users profiles/activities/preferences and, based on this knowledge, may provide a useful, enjoyable service and ensure the quality of the platform.

To address these needs, we developed a Recommender System and a Reporting and Trending Analysis System. The Recommender System (Demo version) is developed to replace the current Random Recommender System of Playnify. The purpose of this system is to recommend users to others users (athletes), that they probably would like to add to their athletes's network, as well as events and venues appropriate to their sports profiles. The Reporting and Trending Analysis System is directed to the administrators of the platform, responsible for monitoring the activities, habits and trends of Playnify's users.

For the development of the Recommender System we used SQL in the implementation of the recommendation algorithms, and ASP.NET for the development of the prototype. Regarding to the Reporting and Trending Analysis System, we first developed reports with SAP Crystal Reports for Microsoft Visual Studio 2010. Then we integrated them with the system web pages, developed in ASP.NET.

The Demo version of the Recommender System will be integrated later with Playnify, which is still in the process of continuos development. The Reporting and Trending Analysis System is already in production environment.



# Índice

<b>CAPÍTULO 1.....</b>	<b>1</b>
<b>INTRODUÇÃO.....</b>	<b>1</b>
1.1.    ENQUADRAMENTO DA EMPRESA .....	1
1.2.    MOTIVAÇÃO.....	2
1.3.    DESCRIÇÃO DO PROBLEMA .....	2
1.4.    ESTRUTURA RELATÓRIO DE ESTÁGIO.....	3
<b>CAPÍTULO 2.....</b>	<b>5</b>
<b>ESTADO DA ARTE .....</b>	<b>5</b>
2.1.    REDES SOCIAIS.....	6
2.2.    MONITORIZAÇÃO DE REDES SOCIAIS .....	10
2.3.    SISTEMAS DE RECOMENDAÇÃO .....	11
2.4.    SISTEMA DE RELATÓRIOS E ANÁLISE DE TENDÊNCIAS.....	13
2.5.    OUTRAS TECNOLOGIAS.....	15
<b>CAPÍTULO 3.....</b>	<b>19</b>
<b>ESPECIFICAÇÃO .....</b>	<b>19</b>
3.1.    DESCRIÇÃO DO PROBLEMA .....	19
3.2.    CARACTERÍSTICAS DOS UTILIZADORES.....	20
3.3.    REQUISITOS DO SISTEMA .....	20
3.4.    CASOS DE USO.....	24
<b>CAPÍTULO 4.....</b>	<b>29</b>
<b>DESENVOLVIMENTO .....</b>	<b>29</b>

4.1.	ARQUITETURA .....	29
4.2.	MODELO DE DADOS .....	32
4.3.	MODELO DE RECOMENDAÇÃO.....	34
4.4.	SISTEMA DE RELATÓRIOS E ANÁLISE DE TENDÊNCIAS.....	40
<b>CAPÍTULO 5 .....</b>		<b>51</b>
<b>TESTES E AVALIAÇÃO .....</b>		<b>51</b>
5.1.	TESTES.....	51
5.2.	AVALIAÇÃO.....	54
<b>CAPÍTULO 6 .....</b>		<b>57</b>
<b>CONCLUSÃO E TRABALHO FUTURO .....</b>		<b>57</b>
6.1.	CONCLUSÃO.....	57
6.2.	TRABALHO FUTURO.....	58
<b>REFERÊNCIAS .....</b>		<b>59</b>
<b>ANEXOS.....</b>		<b>61</b>

# Lista de Tabelas

Tabela 1. User Stories do Sistema de Recomendação .....	21
Tabela 2. User Stories do Sistema de Relatórios e Análise de Tendências.....	22
Tabela 3. Caso de Uso - Autenticação do Sistema de Recomendação (versão Demo). 25	
Tabela 4. Caso de Uso - Recomenda os eventos ao utilizador ativo .....	25
Tabela 5. Caso de Uso - Autenticação do Sistema Relatórios e Análise de Tendências	26
Tabela 6. Caso de Uso - Estatística de nível por desporto de cada região.....	26
Tabela 7. Caso de Uso - Estatística da frequência dos eventos (por desporto) por dia em cada cidade.....	26
Tabela 8. Casos de teste representativos do Sistema de Recomendação. ....	52
Tabela 9. Casos de teste representativos do Sistema de Relatórios e Análise de Tendências .....	53





# Lista de Figuras

Figura 1. Ilustração de redes sociais .....	6
Figura 2. Recomendação do Amazon com base do item comprado pelo cliente. ....	12
Figura 3. Ciclo do AJAX [15] .....	16
Figura 4. Arquitetura lógica .....	30
Figura 5. Arquitetura física .....	31
Figura 6. Modelo relacional .....	33
Figura 7. Amizade dos desportistas .....	35
Figura 8. Diagrama do algoritmo de recomendação dos desportistas .....	36
Figura 9. Diagrama do algoritmo de recomendação dos eventos .....	37
Figura 10. Diagrama do algoritmo de recomendação de instalação desportiva .....	38
Figura 11. Página de autenticação do Sistema de Recomendação (versão Demo) .....	39
Figura 12. Lista de recomendações .....	40
Figura 13. Lista do filtro de país .....	42
Figura 14. Calendário utilizado no filtro de data.....	42
Figura 15. Lista de filtro do desporto .....	43
Figura 16. Filtros do Sistema de Relatórios e Análise de Tendências .....	43
Figura 17. Estatística de Relatórios de skill level dos desportistas por desporto .....	47
Figura 18. Estatística de Análise de Tendências categorizada por dia .....	48
Figura 19. Página de autenticação do Sistema de Relatórios e Análise de Tendências	49
Figura 20. Página principal, menus e submenus do Sistema de Relatórios e Análise de Tendências .....	49



# Abreviaturas e Acrónimos

ADO	ActiveX Data Objects
AJAX	Asynchronous Javascript and XML
ANSI	American National Standards Institute
ASP	Active Server Pages
API	Application Programming Interface
BIRT	Business Intelligence and Reporting Tools
CLR	Common Language Runtime
COM	Component Object Model
CR	Crystal Reports
CSS	Cascading Style Sheet
CSV	Comma-separated Values
DAO	Data Access Object
DLL	Dynamic Link Library
DSN	Data Source Name
DOM	Document Object Model
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IBM	International Business Machines
ID	Identificação
IDE	Integrated Development Enviroment
IIS	Internet Information System
IMS	IP Multimedia Subsystem
Java EE	Java Enterprise Edition
M2M	Machine to Machine
MS SQL	Microsoft SQL
ODA	Open Data Access
ODBC	Open Database Connectivity
OLE DB	Object Linking and Embedding Database
Open Source	O termo que normalmente refere-se a software que conhecido por software livre
R & D	Research and Development
RTF	Rich Text Format
RPT	Extensão do ficheiro de Crystal Report

SAP	Uma empresa de desenvolvimento de software e gestão de empresa
SGDB	Sistema Gestão da Base de Dados
SQL	Structured Query Language
SRSs	Sites de Redes Sociais
TCP/IP	Transmission Control Protocol / Internet Protocol
XML	eXtensible Markup Language

# Capítulo 1

## Introdução

Neste capítulo apresentamos a empresa onde desenvolvemos o projeto de estágio, a motivação para o trabalho, uma descrição dos problemas tratados e os seus objetivos para ajudar perceber melhor o contexto deste projeto. Por fim, são descritos os restantes capítulos deste relatório.

### 1.1. Enquadramento da Empresa

A empresa Ubiwhere foi formada em Setembro de 2007, em Aveiro. A Ubiwhere está localizada no Mirador Business Centre em Aveiro e Sanjotec, Centro Empresarial e Tecnológico de S. J. da Madeira. A Ubiwhere conta com mais de 30 funcionários, na sua maioria desenvolvedores de software. O objetivo principal da empresa é desenvolver tecnologias de ponta. A Ubiwhere atua principalmente em quatro áreas de negócio:

- Telecomunicações;
- Mobilidade;
- Energia;
- Conhecimento e Educação.

Em Telecomunicações, a Ubiwhere fornece serviços de consultoria de R&D e desenvolvimento de software, principalmente em áreas como a qualidade de serviço, cobertura da rede, M2M e plataforma IMS. Em Mobilidade, a Ubiwhere atribui foco especial ao turismo, desenvolvendo aplicações adaptadas ao contexto e baseadas em localização, que enriquecem a experiência de turistas. Em Energia, a Ubiwhere desenvolve soluções para gestão de energia, monitorização e *Data Mining*. Em Conhecimento e Educação, a Ubiwhere desenvolve software para ecrãs interativos e soluções avançadas de colaboração em tempo real.

Um dos produtos desenvolvidos pela Ubiwhere é uma rede social, o Playnify, direcionado para utilizadores desportistas. Foi lançado em final do 2012 e é objetivo deste projeto contribuir para o melhoramento das suas funcionalidades.

## 1.2. Motivação

Em sistemas com muitos utilizadores e globalmente distribuídos torna-se necessária a avaliação da utilização e satisfação dos intervenientes, assim como prever que uma funcionalidade se irá tornar uma chave de sucesso, ou que não está a ter a aceitação prevista. Além de ter um número elevado de utilizadores e conhecer a sua distribuição, é importante perceber como se relacionam os vários tipos de utilizadores, as suas atividades principais, e como tornar ou manter a plataforma interativa, agradável e desejável. Em projetos com um elevado número de utilizadores existe também um igualmente elevado potencial de negócio, pelo que é crucial ter relatórios e dados sobre a sua utilização, indicadores e alertas para assegurar um melhoramento progressivo da qualidade do sistema.

O corrente projeto consiste no desenvolvimento de um Sistema de Recomendação e de um Sistema de Relatórios e Análise de Tendências, com o objetivo de fornecer aos utilizadores da plataforma Playnify um serviço útil e agradável, e aos administradores os relatórios e dados sobre a sua utilização.

## 1.3. Descrição do Problema

A plataforma de rede social, o Playnify, desenvolvida pela Ubiwhere, está neste momento disponível para 250 países e tem mais do que 5000 utilizadores registados. Para melhorar as funcionalidades da plataforma e também poder estudar os perfis dos seus utilizadores (quanto a preferências de desportos, marcas, etc.) desenvolveram-se neste projeto de estágio as seguintes aplicações:

- Um Sistema de Recomendação que integra com o Playnify (para substituir o sistema de recomendação aleatória do Playnify) para identificar melhor os perfis e as atividades dos utilizadores duma maneira inteligente, sem interferir com a sua privacidade;
- Um Sistema de Relatórios e Análise de Tendências para fornecer dados estatísticos e relatórios de utilização, tendências, e hábitos dos utilizadores.

O Playnify possui neste momento um sistema de recomendação aleatória da categoria “desportistas” e “eventos”. Contudo, este sistema de recomendação tem muitas limitações. Recomenda aleatoriamente ao utilizador ativo todos os amigos dos seus

amigos dos quais ainda não é amigo, e todos os eventos abertos e públicos em que o utilizador ativo ainda não se inscreveu.

Para substituir esta funcionalidade, desenvolvemos neste projeto de estágio o Sistema de Recomendação que deverá integrar com o Playnify e fornecer aos utilizadores ativos em tempo real, até dez recomendações nas categorias “Desportistas”, “Eventos” e “Instalações Desportivas”. Estas recomendações são obtidas a partir da análise do perfil dos utilizadores e dos seus amigos.

O Sistema de Relatórios e Análise de Tendências deverá fornecer estatísticas de utilização da plataforma. Identifica o uso do sistema, as tendências e os hábitos baseado em dados históricos registados na base de dados do Playnify. As informações de Relatórios são, sobretudo, compostos por estatísticas sobre os utilizadores (desportistas), desportos, marcas, eventos, e instalações desportivas. As informações de Análise de Tendências são, sobretudo, compostos por estatísticas das análise de tendências dos utilizadores que praticam um ou mais desportos num período de tempo categorizado por mês, semana, dia e hora.

## 1.4. Estrutura Relatório de Estágio

Para além da introdução, o presente relatório de estágio contém mais cinco capítulos. No próximo capítulo é apresentado o Estado da Arte relativo a Redes Sociais, Monitorização de Redes Sociais, Sistemas de Recomendação, Sistema de Relatórios e Análise de Tendências, assim como as ferramentas utilizadas na implementação da solução final. No capítulo 3 é apresentada uma descrição mais detalhada do problema. O capítulo 4 contém uma descrição das soluções desenvolvidas, incluindo a descrição de arquitetura do sistema, modelos de dados e sua implementação. No capítulo 5 apresentam-se testes e uma avaliação dos sistemas desenvolvidos neste projeto de estágio. Finalmente, no capítulo 6 são apresentadas algumas conclusões, assim como algumas linhas possíveis para trabalho futuro após a conclusão deste projeto de estágio.





# Capítulo 2

## Estado da Arte

Neste capítulo fazemos uma breve revisão de alguns aspetos relacionados com este projeto de estágio. Os assuntos abordados são nomeadamente:

- Redes Sociais;
- Monitorização de Redes Sociais;
- Sistemas de Recomendação;
- Sistemas de Relatórios e Análise de Tendências;
- Outras Tecnologias Utilizadas.

Uma vez que o projeto é baseado numa rede social, começamos por fazer uma breve descrição de algumas redes sociais com o objetivo de procurarmos compreender as funcionalidades, evoluções, e serviços fornecidos aos utilizadores.

Na secção de Monitorização de Redes Sociais estudamos alguns tipos de monitorização de redes sociais que se aplicam neste projeto de estágio.

Na secção de Sistemas de Recomendação estudamos algumas técnicas de recomendação existentes que se relacionam com o Sistema de Recomendação, desenvolvido durante este estágio.

Na secção de Sistemas Relatórios e Análise de Tendências estudamos alguns produtos/ferramentas existentes usados para desenvolver o Sistema de Relatórios e Análise de Tendências.

Finalmente na secção de Outras Tecnologias apresentamos outras tecnologias além das ferramentas abordadas na secção anterior, que foram utilizadas neste projeto de estágio.

## 2.1. Redes Sociais

Os primeiros sites de redes sociais surgiram já nos anos 90, mas foi durante as duas últimas décadas que os sites de redes sociais (SRSs) têm adquirido cada vez mais importância, sendo hoje em dia utilizados por milhões de pessoas. Algumas redes sociais surgem exatamente para responder às necessidades das pessoas partilharem conteúdos com os outros utilizadores com um interesse comum, por exemplo, o gosto pelo desporto, a fotografia ou a arte.

Boid & Ellison [1] definiu os SRS como 1) Serviços baseados na *web* que permitem a um indivíduo criar um perfil público ou semi-público dentro de um sistema limitado; 2) Articulação da lista de cada utilizador com as de outros com quem partilha conexão; e 3) Consulta da lista do seus relacionamentos na rede e dos relacionamentos destes na mesma rede.



Figura 1. Ilustração de redes sociais

Existe atualmente um número considerável de redes sociais com diferentes funcionalidades e um público alvo diversificado. Seleccionámos algumas entre as que têm maior grau de utilização (Figura 1), estudámo-las e apresentamos a seguir as suas descrições sumárias.

### Friendster

O Friendster<sup>1</sup> surgiu em 2002 e inicialmente tinha funcionalidades parecidas com outras redes sociais, como o Facebook, descrito mais à frente. É sobretudo popular no sudeste da Ásia [2].

---

<sup>1</sup> [www.friendster.com](http://www.friendster.com)

Devido ao crescimento do Facebook, o tráfego deste site sofreu uma grande queda e só voltou a ganhar popularidade depois de, em 2011 [3], ser relançado como site especializado em jogos sociais.

## MySpace

O MySpace<sup>2</sup> surgiu em 2003 como site de rede social especializado em música e fotografia. Em 2009, foi redesenhado para fazer face à diminuição do número de utilizadores que tinha vindo a sofrer. Porém, esta estratégia não contrariou a tendência do site perder popularidade e utilizadores. Em 2012 o MySpace surgiu redesenhado e com novo nome “New MySpace<sup>3</sup>”, com uma funcionalidade de pesquisa simples, sobretudo, para facilitar a pesquisa de músicas e vídeos.

## LinkedIn

O LinkedIn<sup>4</sup> é um site de rede social para profissionais que surgiu em 2003. O principal propósito do site é permitir ao utilizador partilhar as informações profissionais, procurar/encontrar emprego e oportunidades de negócio. O LinkedIn permite também aos utilizadores endossarem competências de outros membros da rede e formarem grupos de interesse específicos.

O LinkedIn tem um *plug-in* para empresas colocarem listas de vagas de emprego para os utilizadores poderem concorrer às mesmas através do próprio LinkedIn.

## Orkut

O Orkut<sup>5</sup> é um site de rede social baseado em amizades que era muito popular no Brasil e na Índia. O Orkut disponibiliza aos utilizadores uma ferramenta de *chat* integrada que aparece automaticamente no rodapé da página, tal como no Facebook.

Em 2009, o Orkut lançou o “Novo Orkut” com um *design* totalmente reformulado para oferecer recursos mais modernos aos utilizadores. Contudo, o site perdeu o seu espaço no mercado em 2011 [4] enquanto outras redes sociais como o Facebook e o Twitter têm crescido cada vez mais.

---

<sup>2</sup> [www.myspace.com](http://www.myspace.com)

<sup>3</sup> <https://new.myspace.com>

<sup>4</sup> [www.linkedin.com](http://www.linkedin.com)

<sup>5</sup> [www.orkut.com](http://www.orkut.com)

## Facebook

O Facebook<sup>6</sup> é o site da rede social com maior grau de utilização até a data. Fundado por Mark Zuckerberg, foi lançado em 2004. Facebook permite aos seus utilizadores interagir com outros através de publicações no mural, mensagens, *chat*, comentários a publicações, participação em jogos e em grupos, audição partilhada de música, etc. Além disso, o Facebook também tem recursos para *marketing*, como o *Marketplace*.

## Twitter

O Twitter<sup>7</sup> é um site de rede social de *microblogging* que surgiu em 2006 e permite ao utilizador enviar e receber notificações de outros contactos por meio do *website* e por SMS. As notificações são textos até 140 caracteres, conhecido como “*tweets*”.

Um utilizador pode interagir como os outros através do *retweet*, que é análogo aos comentários da publicação do Facebook. A ferramenta *trending topics* do Twitter é uma lista, em tempo real, das frases mais publicadas no Twitter, sendo por omissão dez frases. Valem para essa lista as *hashtags* (#) e nomes próprios.

## Foursquare

O Foursquare<sup>8</sup> é um site de rede social baseado na localização geográfica, que surgiu em 2009. O site permite ao utilizador publicar a sua presença num local, chamado “*Check in*”, e encontrar os amigos que estejam mais próximos da sua localização.

O Foursquare permite ainda ao utilizador procurar locais específicos próximos da sua localização geográfica, como por exemplo, uma instalação desportiva, uma rede sem fios gratuita, restaurantes e outros.

## Pinterest

O Pinterest<sup>9</sup> é um site de rede social que surgiu em 2009 e foi lançado em 2010 para partilha de fotografias. O site permite carregar, guardar, classificar e gerir imagens, conhecidas como “*pins*”. A maior parte dos utilizadores são mulheres e é muito popular nos Estados Unidos .

---

<sup>6</sup> [www.facebook.com](http://www.facebook.com)

<sup>7</sup> [www.twitter.com](http://www.twitter.com)

<sup>8</sup> [www.foursquare.com](http://www.foursquare.com)

<sup>9</sup> [www.pinterest.com](http://www.pinterest.com)

Um ano depois do seu lançamento, o Pinterest tornou-se um dos dez maiores serviços de rede social com 11 milhões de visitas por semanas [5]. Observamos que uma das razões de crescimento do Pinterest talvez seja o facto do Pinterest ser um lugar de descoberta, onde é fácil encontrar produtos de diferentes categorias.

### Google+

O Google+<sup>10</sup> é o site de rede social do *Google Inc*, parecido com o Facebook, que surgiu em 2011. Além de recursos semelhantes aos do Facebook, o Google+ possui um recurso específico - o *hangout*, que facilita conversas de vídeo em grupo, de no máximo dez pessoas.

O Google+ também permite a importação de fotografias do Orkut, a interação com o Youtube e oferece uma lista de tópicos mais populares (*trending topics*).

### Playnify

O Playnify<sup>11</sup> é uma rede social desportiva lançada em finais de 2012. Os desportos disponíveis na plataforma até a data são *Basketball, Cycling, Futsal, Handball, Mountain Bike, Padel, Rugby, Running, Soccer 7, Soccer 11, Squash, Table Tennis, Tennis e Volleyball*. Esta rede permite aos seus utilizadores (desportistas) registar e definir as suas preferências de desporto com o respetivo nível (*skill level*), as disponibilidades para jogar e as marcas dos seus equipamentos preferidos.

O Playnify permite criar equipas, eventos e inserir/registar novas instalações desportivas. Através do Playnify, os utilizadores podem organizar eventos em determinadas instalações desportivas e convidar amigos para esses eventos.

O Playnify também permite convidar os amigos através do Facebook, Gmail ou por email. Permite ao utilizador comunicar com outros amigos através de mensagens e deixar comentários aos eventos a que se junta. A atualização de todas as atividades é acompanhada através das notificações.

### Resumo

Alguns sites de redes sociais, por exemplo, o Friendster, o MySpace e o Orkut, foram perdendo atividade devido ao crescimento de outros, especialmente o Facebook e o

---

<sup>10</sup> <https://plus.google.com>

<sup>11</sup> [www.playnify.com](http://www.playnify.com)

Twitter. O LinkedIn continua a crescer por ser um site de rede social com objetivos e funcionalidades específicas que diferem das do Facebook e do Twitter. O Google+ surgiu com funcionalidades semelhantes às do Facebook, contudo não há nenhuma indicação de que os utilizadores do Facebook tenham migrado para este novo SRS do Google. O Facebook é o site de rede social com maior número de utilizadores até ao presente.

O Foursquare surgiu com uma nova funcionalidade baseada em localização geográfica e tornou-se uma rede de aplicação para equipamentos móveis. É mais usada para procurar locais próximos da localização geográfica do utilizador, como por exemplo restaurantes.

Pinterest com as funcionalidades atrativas e facilidade de utilização surgiu como um lugar de descoberta que ganhou um número elevado dos utilizadores apenas em dois anos. E em 2012, Playnify surgiu como um site de rede social especializado em desporto, que é bastante diferente dos restantes.

De facto, parece que hoje em dia, para poder garantir um tráfego regular e assim ser competitivo a nível comercial e de *marketing*, uma rede social deve ter um tema específico que interesse a um grupo alvo de utilizadores, e aos quais deve dar o maior apoio possível. É neste contexto que a monitorização se torna importante.

## 2.2. Monitorização de Redes Sociais

Nos parágrafos seguintes descrevemos dois tipos de monitorização de redes sociais que são usados para estudar o crescimento do uso, os comportamentos, hábitos, e as tendências dos utilizadores ao longo do tempo.

### Estatísticas

Para ter uma boa monitorização de uma rede social é importante ter estatísticas sobre o número de utilizadores, categorizadas por localização (país/região/cidade) e por períodos (ano/mes/semana/dia). Uma das ferramentas *online* do Google para apresentação de estatísticas de redes sociais é o *Google Analytic*<sup>12</sup> que é gratuita, e serve principalmente para apresentar a estatística de visitação dos utilizadores ao site.

---

<sup>12</sup> [www.google.com/analytics](http://www.google.com/analytics)

Neste projeto, desenvolvemos estatísticas do uso do Playnify. As estatísticas apresentam informações em gráficos e tabelas cruzadas. As informações são agrupadas por “Desporto”, “Eventos”, “Sexo”, “Região”, etc, e filtradas sobretudo, por “País”, “Região” e “Cidade”. Estas estatísticas são obtidas com o objetivo de fornecer informações do uso da rede aos adiministradores da plataforma Playnify.

### **Análise de Tendências**

No contexto de redes sociais, é muito importante analisar as atividades dos utilizadores para determinar as suas tendências e hábitos. No contexto de *Text Mining*, diversos estudos foram feitos para detectar as tendências dos utilizadores tanto *offline* como *online*, utilizando diferentes métodos, medidas ou regras. Uma delas é a medida *tf-idf* (*term frequency, inverse document frequency*) que foi usada em vários estudos como os de, Cvijikj & Michahelles [6], que a usaram para identificar os tópicos mais frequentes das publicações no Facebook; ou Uchida & Shibata [7], que também usaram para identificar os tópicos mais discutidos na comunidade da blogosfera. Por outro lado, F. Fang et al [8], usaram a medida de *support* para identificar o grupo das palavras mais frequentes do Twitter (*tweets*).

Neste projeto de estágio, o problema da deteção de tendências é um problema de *Data Mining*. Um trabalho semelhante ao nosso problema é a análise da duração de atividades de *Check-In* no Foursquare descrita em [9], onde as tendências são identificadas com base na hora *Check-In* dos utilizadores. Seguindo a lógica deste trabalho, desenvolvemos o Sistema de Análise de Tendências.

### **Resumo**

Neste projeto, o sistema de monitorização do Playnify é composto por monitorização do uso e das tendências com base na localização. Estas informações são apresentadas através de gráficos e de tabelas cruzadas nos relatórios do sistema.

## **2.3. Sistemas de Recomendação**

No contexto das redes sociais, os sistemas de recomendação analisam os perfis dos utilizadores, as suas amizades e os itens alvo para ajudá-los a encontrar os itens de acordo com os seus interesses. Existem dois tipos de sistemas de recomendação: *content-based* e *collaborative filtering algorithms*.

Nos *content-based algorithms* a escolha das recomendações a um utilizador é baseada no seu perfil [10]. Por exemplo, para um desportista que pratica basquetebol, um sistema de recomendação *content-based* recomendaria listas de eventos de basquetebol. Outro exemplo é o *website* do Google, onde um algoritmo *content-based* fornece informação (por exemplo notícias) ao utilizador baseado na sua localização atual.

Nos *collaborative filtering algorithms* a escolha das recomendações é baseada no histórico de todos os utilizadores [10], tendo também em conta transações e *ratings* de produtos, etc.

Uma das empresas que utiliza um sistema de recomendação como ferramenta principal na área de *marketing online* é o Amazon. Uma das técnicas de recomendação do Amazon descrita em [11] é o *Item-to-Item Collaborative Filtering*. O algoritmo referido gera a lista de recomendações para um cliente com itens semelhantes aos que ele selecionou (Figura 2). A semelhança de itens mede-se pela quantidade de clientes que os compraram juntos. Isto é, uma recomendação baseada na similaridade dos itens e não dos utilizadores.



Figura 2. Recomendação do Amazon com base do item comprado pelo cliente.

No nosso problema concreto, a lista de recomendação é gerada baseada não apenas no perfil e localização do utilizador, como também em dados históricos de amizade



registados na base de dados. Assim é um problema que se deve resolver utilizando os dois, *content-based* e *collaborative filtering algorithms*.

Como na nossa pesquisa não encontrámos nenhum algoritmo que pudesse ser diretamente aplicado ao nosso problema, desenvolvemos neste projeto de estágio, um novo algoritmo de recomendação que é usado no Sistema de Recomendação versão Demo para o Playnify.

## 2.4. Sistema de Relatórios e Análise de Tendências

Nesta secção descrevemos alguns produtos para desenvolver Sistemas de Relatórios e Análise de Tendências, que foram analisados no âmbito deste projeto. O objetivo desta análise foi identificar os produtos que pudessem ser usados com mais vantagens.

### Crystal Reports

O Crystal Reports (CR)<sup>13</sup> é uma aplicação de *business intelligence*, específica para a criação de relatórios a partir de uma vasta gama de fontes de dados (MS SQL, MySQL, PostgreSQL, Oracle, e os outros). O CR comunica com várias fontes de dados através de *data providers* como, por exemplo, OLE DB, ODBC, DAO, e COM.

O CR permite desenhar, visualizar e formatar relatórios. Para tal possui funcionalidades para a gestão de listas, agrupamento de dados, tabelas, tabelas cruzadas, gráficos e sub-relatórios. Os utilizadores finais podem explorar ou filtrar os dados que são apresentados no relatório através da definição de parâmetros. O relatório é integrado com opções flexíveis de exportação para formatos pdf, excel, html, xml, RTF, etc.

O CR possui funcionalidade para entrega de relatórios através da web e e-mail do Microsoft Office. Inclui ainda funcionalidade limitada para sistemas de informação geográfica. A versão atual do Crystal Reports é a Crystal Reports 2011 do SAP.

### BIRT

O *Business Intelligence and Reporting Tools* (BIRT)<sup>14</sup> é um sistema de *reporting* do Eclipse para aplicações *web* que se baseia em Java e Java EE. É *open source*, composto por um *designer* de relatório e um *runtime* para o servidor da aplicação.

---

<sup>13</sup> [www.crystalreports.com](http://www.crystalreports.com)

<sup>14</sup> [www.eclipse.org/birt](http://www.eclipse.org/birt)

O BIRT utiliza o *Open Data Access* (ODA) para comunicar com as fontes de dados e permite combinar diferentes fontes de dados usando *inner* e *outer join*. *Inner* e *Outer Join* são operações do SQL que realizam a junção de duas ou mais tabelas. Para converter os dados brutos em informações úteis (relatórios) utiliza JavaScript.

O BIRT fornece listas, gráficos, tabelas, tabelas cruzadas, textos e documentos, e relatórios compostos. Os relatórios apresentam os dados ordenados, resumidos, filtrados, e agrupados em totais e percentagem.

## JReport

O JReport<sup>15</sup> é uma aplicação *business intelligence* baseado do Java. Fornece uma visualização de dados avançada com um sistema de painel poderoso e uma solução para dispositivos móveis.

O JReport possui mais de 40 tipos de gráficos e estilos, tabelas, tabelas cruzadas e permite também a visualização do Google Map. Reforçado com gráficos, incluindo animação, permite mostrar a tendência de dados. O JReport permite exportar os relatórios para pdf, excel, html, xml e csv, e permite o envio do relatório via email.

## Scriptcase

O Scriptcase<sup>16</sup> é uma ferramenta para desenvolvimento web baseado no PHP; comunica com as bases de dados MS SQL, MySQL, PostgreSQL, Oracle e outros através de ODBC, ADO e COM.

O Scriptcase permite criar um relatório dinâmico que oferece ao utilizador final agrupamento de dados, tabelas cruzadas, cálculo de totais, e gráficos em flash. Permite criar filtros, ordenação, pesquisa, integração com Google Maps e exportação para formatos, como xls, xml e pdf.

## Resumo

Depois de estudar os diferentes produtos/ferramentas, serviços associados e soluções elaboradas por cada um deles, notamos que em termos de Relatórios e Análise de Tendências, as soluções de todos são bastante semelhantes.

---

<sup>15</sup> [www.jinfony.com](http://www.jinfony.com)

<sup>16</sup> [www.scriptcase.com.br](http://www.scriptcase.com.br)

No entanto optou-se neste projeto pelo Crystal Reports, uma vez que tem uma versão gratuita que integra com o ambiente de desenvolvimento integrado Microsoft Visual Studio 2010, já adquirido pela empresa. Por outro lado, contou também a compatibilidade dos recursos – ASP.NET com Crystal Reports através do Crystal Reports Viewer (o componente .NET do CR), a compatibilidade do NPQSQL com ASP.NET, e a minha experiência de trabalho com Crystal Reports.

## 2.5. Outras Tecnologias

Nesta secção apresentamos as outras tecnologias, além das ferramentas indicadas na Secção 2.4, utilizadas no desenvolvimento dos nossos Sistemas de Recomendação e de Relatórios e Análise de Tendências.

### **Microsoft Visual Studio 2010**

Como ambiente de desenvolvimento da aplicação, usamos o Microsoft Visual Studio é um ambiente de desenvolvimento integrado (IDE) da Microsoft, composto por um pacote de programas para desenvolvimento de software. O Visual Studio suporta diferentes linguagens de programação, dedicando-se sobretudo ao .NET *Framework* e às linguagens Visual Basic, C, C++, C# e J#. É também um grande IDE de desenvolvimento de aplicações para a *web* usando a plataforma ASP.NET. As linguagens mais utilizadas nesta plataforma são o VB.NET e o C#.

O Visual Studio apresenta várias edições das quais se destacam o Visual Studio Ultimate, o Express, Premium e o Professional. A versão atual do Visual Studio é Visual Studio 2012 [12], contudo, neste trabalho usamos o Visual Studio 2010 pelo facto da empresa possuir esta versão.

### **SAP Crystal Reports Developer Version for Microsoft Visual Studio 2010**

No desenho de relatórios, usamos o *SAP Crystal Reports Developer* para Microsoft Visual Studio, que é um software gratuito do SAP para o desenho de relatórios que permite a integração com o Visual Studio. A versão 2010 (Crystal Reports 10) usada no projeto foi desenvolvida para o Visual Studio 2010. Possui praticamente as mesmas funcionalidades do SAP Crystal Reports 2011. A diferença mais significativa reside no facto da versão gratuita apenas trabalhar com os relatórios desenvolvidos na versão atual, enquanto que o SAP Crystal Reports 2011 para Visual Studio é compatível com versões anteriores [13].

## Microsoft .NET Framework

.NET Framework é uma das plataformas de desenvolvimento mais populares para a construção de aplicações para Windows. A plataforma .NET inclui linguagens de programação como C#, Visual Basic, Common Language Runtime (CLR) e bibliotecas de classes. CLR é responsável pela interface entre o código e o sistema operacional. Neste projeto utilizamos ASP.NET *Web Application* para definir os campos de filtros e os protótipos, e a linguagem C# como *code behind* que definem eventos de filtragem e as rotinas para integrar com os relatórios.

## HTML 5

O HTML (*HyperText Markup Language*) é uma linguagem para a apresentação de conteúdos utilizada para criação de páginas web e outro tipo de documentos possíveis de visualizar utilizando um browser [14]. Neste projeto utilizamos a versão HTML5 por ser a versão mais recente (ainda em desenvolvimento [15]), e também por ser compatível com o Visual Studio 2010.

## CSS 3

CSS (*Cascading Style Sheet*) é uma linguagem de estilos de página usada para estruturar a apresentação de um documento HTML. Tal como o HTML5 a versão CSS3 ainda se encontra em fase de desenvolvimento [16].

## AJAX

AJAX (*Asynchronous JavaScript and XML*) é uma técnica para atualizar partes de páginas web, sem recarregar a página inteira. AJAX permite que uma aplicação web envie e receba dados do servidor de forma assíncrona sem interferir com a exibição e o comportamento da página existente.

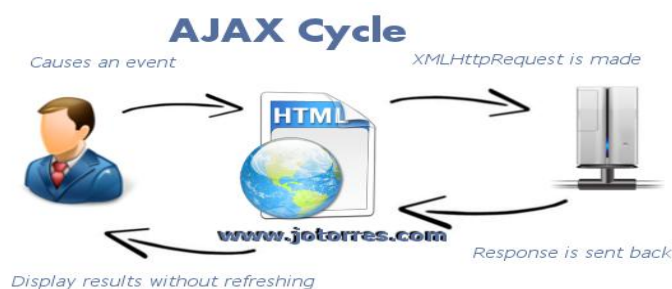


Figura 3. Ciclo do AJAX [15]

AJAX pode ser utilizado com:

- XMLHttpRequest object para enviar/receber dados de forma assíncrona com o servidor;
- JavaScript/DOM para mostrar/interagir com informação;
- CSS para fornecer estilos de dados;
- XML para formatar os dados transferidos.

## SQL

O SQL (*Structured Query Language*) é uma linguagem para armazenar, manipular e recuperar dados armazenados em bases de dados relacionais. Sistemas como MySQL, Microsoft Access, Oracle, IBM DB2, PostgreSQL e Microsoft SQL Server usam SQL como linguagem padrão da base de dados.

## PostgreSQL

Como sistema de gestão da nossa base de dados, usamos o PostgreSQL que é um sistema de dados relacionais de código aberto. Utiliza o SQL como linguagem padrão da base de dados.

## NPGSQL

NPSQL é um *data provider* do .NET para servidor da base de dados do PostgreSQL [18]. Permite uma aplicação cliente .NET (ASP.NET, Web Services, etc.) para enviar e receber dados do servidor de PostgreSQL.

A utilização do NPGSQL em aplicações .NET é semelhante com acesso de dados da base de dados através do OLE DB. Em C#, adiciona diretamente para a página apropriada ou classe.

## Object Linking and Embedding Database (OLE DB)

OLE DB é um API desenhado pela Microsoft para permitir acesso a bases de dados e a manipulação de dados a nível do sistema operativo. *ActiveX Data Object* (ADO) utiliza OLE DB para comunicar com OLE DB *providers* e obter os dados.

Um componente do OLE DB é o *data provider* que fornece dados. O OLE DB tem um caminho para o ODBC (*Microsoft OLE DB provider for ODBC Drivers* que utiliza neste

projeto) de forma a permitir que sejam reutilizadas todas as funcionalidades dos *drivers* de ODBC.

### Open Database Connectivity (ODBC)

ODBC é um API para aceder ao sistema de gestão da base de dados (SGDB), que suporta vários tipos de SGDB, entre os quais, o PostgreSQL.

O psqLODBC ANSI é um *driver* ODBC para PostgreSQL fornecido para Windows, que suporta aplicações através do ANSI ODBC API.

### Resumo

Resumindo, utilizamos neste projeto:

- O Visual Studio 2010 como ambiente de desenvolvimento integrado;
- Uma versão gratuita do SAP Crystal Reports para Visual Studio, para desenhar e visualizar relatórios;
- NPGSQL é utilizado juntamente com ASP.NET (C#) para conectar com a base de dados PostgreSQL;
- Microsoft OLE DB provider for ODBC Drivers é utilizado pelo Crystal Reports para conectar com a base de dados PostgreSQL;
- SQL para consultar e extrair informação da base de dados;
- HTML5 e CSS3 para definir a parte de interface e filtros;
- AJAX para fazer atualização direta dos campos de filtros.

# Capítulo 3

## Especificação

Neste capítulo apresentamos uma descrição detalhada do problema abordado neste projeto, caracterizamos os tipos de utilizadores alvo, e especificamos os requisitos a que o sistema deve obedecer.

### 3.1. Descrição do Problema

O presente projeto de estágio consiste no desenvolvimento de duas aplicações para a plataforma Playnify: um Sistema de Recomendação, que identifica os perfis e as atividades dos utilizadores duma maneira inteligente; e um Sistemas de Relatórios e Análise de Tendências, que fornece dados estatísticos e relatórios de utilização, tendências, e hábitos dos utilizadores.

O Sistema de Recomendação é desenvolvido para integrar com o Playnify e fornece até dez recomendações para os utilizadores ativos com base nos seus perfis e nos perfis dos seus amigos. As recomendações são geradas para cada uma das seguintes categorias de recomendação:

- *Desportistas*: recomenda amigos dos amigos do utilizador ativo que não são amigos dele, aos quais ainda não enviou um pedido de amizade, que moram na mesma cidade, praticam os mesmos desportos com o mesmo nível e que têm as mesmas disponibilidades marcadas na plataforma.
- *Eventos*: recomenda eventos, abertos para inscrição, que são públicos, nos quais o utilizador ativo ainda não se inscreveu, que estão perto (na cidade em que vive), de acordo com a sua disponibilidade, do desporto que pratica, e nos quais alguns desportistas já se inscreveram.
- *Instalação desportiva*: recomenda instalações que estão perto (na cidade onde o utilizador ativo vive) e que ainda não marcou como favoritos, que permitem praticar o tipo de desporto que ele pratica e que alguns desportistas já utilizaram.

O Sistema de Relatórios e Análise de Tendências é um novo sistema independente, desenvolvido para, a partir da base de dados do Playnify, permitir aos administradores do Playnify conhecer o uso da rede, os comportamentos e as tendências dos utilizadores. O Sistema de Relatórios fornece os seguintes tipos de informação:

Informação sobre os utilizadores (desportistas);

- Informação sobre os desportos;
- Informação sobre as marcas dos equipamentos;
- Informação sobre os eventos;
- Informações sobre as instalações desportivas.

A Análise de Tendências fornece relatórios e estatísticas de tendências dos utilizadores que praticam um ou vários desportos num determinado local dentro dum determinado período de tempo. O período é categorizado por mês, semana, dia e hora.

As informações de Relatórios são filtradas por “País”, “Região”, “Cidade” e “Desporto”; e as de Tendências por “País”, “Região”, “Cidade”, “Desporto” e “Intervalo de Tempo”.

## 3.2. Características dos Utilizadores

O Sistema de Recomendação é dirigido aos utilizadores ativos do Playnify aos quais o sistema irá fornecer funcionalidades para encontrar desportistas, eventos e instalações desportivas mais próximas.

O Sistema de Relatórios e Análise de Tendências, é dirigido aos administradores do Playnify aos quais fornece dados sobre o uso da rede, os hábitos e as tendências dos utilizadores.

## 3.3. Requisitos do Sistema

Nesta subsecção apresentamos os requisitos funcionais e não funcionais aos quais o Sistema de Recomendação e o Sistema de Relatórios e Análise de Tendências devem obedecer.



## Requisitos Funcionais

Os requisitos funcionais definem ações que o sistema deve ser capaz de realizar [19]. Os requisitos funcionais são apresentados e detalhados através de *User Stories*. Os *User Stories* entendem-se como os requisitos que são expressos como cenários [20], ou seja, descrições simples que descrevem uma funcionalidade do ponto de vista dos utilizadores. Os *User Stories* são normalmente compostos por três partes: **QUEM** executa a ação, o quê **PRETENDE** fazer, e qual o **OBJETIVO** da ação.

Em seguida, na Tabela 1, apresentamos os *User Stories* do Sistema de Recomendação e na Tabela 2, apresentamos os *User Stories* do Sistema de Relatórios e Análise de Tendências.

Tabela 1. User Stories do Sistema de Recomendação

ID	Quem	Pretende	Objetivo
USR001	Utilizador	Validar o acesso	Permitir o acesso aos utilizadores do Playnify
USR002	Sistema	Encontrar os amigos dos amigos do utilizador ativo, dos quais ainda não é amigo, aos quais ainda não enviou um pedido de amizade, que moram na mesma cidade, praticam pelo menos um mesmo desporto com o mesmo nível, e têm a mesma disponibilidade marcada na plataforma	Recomendar ao utilizador ativo até dez amigos (dos seus amigos), que muito provavelmente ele gostaria de conhecer para praticarem desporto
USR003	Sistema	Encontrar os eventos, abertos para inscrição, dos desportos que o utilizador ativo pratica, na cidade onde ele vive, de acordo com as suas disponibilidades, nas quais alguns desportistas já são inscritos	Recomendar ao utilizador até dez eventos, ordenados por data de início da sua realização, nos quais participam muitos dos desportistas, e nos quais muito provavelmente ele também gostaria de participar
USR004	Sistema	Encontrar as instalações desportivas que estão na cidade onde o utilizador ativo vive, que ainda não utilizou, que ele ainda	Recomendar ao utilizador ativo até dez instalações desportivas mais utilizadas pelos desportistas, onde muito

		não marcou como favoritos, que permitem praticar desportos que ele pratica, e que alguns desportistas já utilizaram	provavelmente ele também gostaria de utilizar para praticar desporto
--	--	---	--

Tabela 2. User Stories do Sistema de Relatórios e Análise de Tendências

ID	Quem	Pretende	Objetivo
USRT001	Utilizador	Validar o acesso	Permitir o acesso apenas aos utilizadores com nível de acesso administrador
USRT002	Utilizador	Estatística total dos desportistas que praticam um desporto	Saber quantos desportistas praticam um desporto; em cada país, região e cidade
USRT003	Utilizador	Estatística total dos desportistas que praticam um desporto categorizado por sexo	Saber quantos desportistas (agrupado por sexo) praticam um desporto; em cada país, região e cidade
USRT004	Utilizador	Estatística total de instalações desportivas agrupado por regiões/cidades	Saber quantas instalações existem; em cada região por país, e cada cidade por região
USRT005	Utilizador	Estatística total de instalações desportivas onde se pratica um determinado desporto	Saber em quantas instalações desportivas é praticado um determinado desporto; em cada país, região e cidade
USRT006	Utilizador	Marcas de equipamento utilizadas por desporto	Saber quais as marcas de equipamento mais utilizadas; por desporto (ou em todos os desportos); em cada país, região e cidade
USRT007	Utilizador	Estatística total dos eventos por desporto em cada instalação desportiva	Saber quantos eventos de cada desporto decorreram numa instalação desportiva; em cada país, região e cidade
USRT008	Utilizador	Estatísticas total dos eventos em cada instalação desportiva	Saber quantos eventos decorreram em cada instalação desportiva; de cada país, região

			e cidade, e por intervalo de datas
USRT009	Utilizador	Estatísticas nível por desporto	Saber qual é o nível que os desportistas de cada desporto possuem; em cada país, região e cidade
USRT010	Utilizador	Estatísticas de instalações favoritas	Conhecer as, até vinte, instalações favoritas em cada país, região e cidade
USRT011	Utilizador	Estatísticas da frequência de eventos por mês/semana/dia/hora dentro dum intervalo de tempo	Conhecer a tendência dos desportistas praticarem um ou mais desportos num período (em que mês/semana/dia/hora os desportistas tendem a praticar um desporto); em cada país, região e cidade, dentro dum intervalo de tempo, e por desportos (ou em todos os desportos)
USRT012	Utilizador	Estatísticas dimensão da rede dos desportistas	Saber quantos utilizadores em cada país, região e cidade têm um certo número de amigos
USRT013	Utilizador	Estatísticas número de mensagens enviadas/recebidas na plataforma	Saber quantos utilizadores em cada país, região e cidade receberam/enviaram um certo de número de mensagem

### Requisitos Não Funcionais

Requisitos não funcionais são restrições sobre os serviços ou funções oferecidas pelo sistema [20]. Trata-se de um tipo de requisito que especifica critérios que são utilizados para avaliar o funcionamento do sistema, ao contrário dos requisitos funcionais que especificam comportamentos específicos que o sistema deve cumprir. Note-se que, os IDs utilizados na identificação dos requisitos não funcionais em seguida, correspondem aos IDs atribuídos acima.

### *Usabilidade*

[ID: USRT014] O Sistema de Relatórios e Análise de Tendências deverá possuir uma utilização intuitiva de maneira que as funcionalidades principais, como a visualização de relatórios (estatísticas) na Tabela 2, possam ser facilmente utilizadas.

### *Fiabilidade*

[ID: USR005] O Sistema de Recomendação deve garantir a integridade com o Playnify de maneira que as gerações automáticas de recomendação sejam geradas sem falhas.

[ID: USRT015] Da mesma maneira, o Sistema de Relatórios e Análise de Tendências deve garantir a integridade com a base de dados do Playnify sem falhas e apresentar os relatórios com dados atualizados, sem perda de informação, conforme a informação na base de dados.

### *Desempenho*

[ID: USR006] O Sistema de Recomendação deve gerar as recomendações em tempo real, com as recomendações atualizadas para qualquer tipo de utilizador.

[ID: USRT016] Um relatório do Sistema de Relatórios e Análise de Tendências deve ser gerado em menos de 30 segundos.

## 3.4. Casos de Uso

Casos de uso descrevem uma sequência de interações entre um ator e um sistema, permitindo ao ator alcançar determinado objetivo [21]. No nosso caso, os atores são utilizadores caracterizados na Secção 3.2, e os sistemas são o Sistema de Recomendação e o Sistema de Relatórios e Análise de Tendências.

Segundo Cockburn [21], a escolha do texto simples para casos de uso é uma boa escolha, embora eles possam ser escritos usando fluxogramas, diagramas de sequências, redes de Petri e linguagens de programação. Com base nesta teoria, escolhemos a forma de texto para os casos de uso do nosso projeto com as seguintes descrições:

- Identificador do caso de uso;
- Nome do caso de uso;

- Descrição sumária;
- Ator(es) que executa(m) a ação: os utilizadores (caracterizados na secção 3.2);
- Pré-condições: a condição inicial para que o caso de uso seja executado;
- Pós-condições: a condição que se deve verificar no final da execução;
- Sequência dos eventos: eventos necessários para executar as ações dadas.

A seguir, apresentamos alguns casos de uso que consideramos representativos. Para cada caso de uso é feita a descrição individual com os detalhes da sequência dos eventos:

Tabela 3. Caso de Uso - Autenticação do Sistema de Recomendação (versão Demo)

Identificador	USR001_UC001
Nome	Autenticação no Sistema de Recomendação (versão Demo)
Descrição sumária	Autenticação dos utilizadores no Sistema de Recomendação
Ator(es)	Utilizador
Pré-condições	-
Pós condições	O utilizador acede à página inicial do Sistema de Recomendação
Sequência de eventos	1. O utilizador introduz o nome de utilizador e palavra-chave <sup>17</sup> e autentica no Sistema de Recomendação 2. Validar o nome do utilizador na base de dados do Playnify 3. O utilizador acede a página do Sistema de Recomendação

Tabela 4. Caso de Uso - Recomenda os eventos ao utilizador ativo

Identificador	USR003_UC002
Nome	Recomendação dos eventos ao utilizador ativo
Descrição sumária	Recomendação ao utilizador de (até) dez eventos na sua cidade em que alguns desportistas já estão inscritos
Ator(es)	Utilizador
Pré-condições	O utilizador é autenticado no Sistema de Recomendação
Pós condições	Listar até dez eventos

<sup>17</sup> Como é uma versão Demo, não fazemos nenhuma validação para a palavra-chave.

Sequência de eventos	Recomendação de uma lista de (até) dez eventos
----------------------	--

Tabela 5. Caso de Uso - Autenticação do Sistema Relatórios e Análise de Tendências

Identificador	USRT001_UC003
Nome	Autenticação no Sistema Relatórios e Análise de Tendências
Descrição sumária	Validação dos utilizadores administradores
Ator(es)	Utilizador
Pré-condições	-
Pós-condições	O utilizador acede à página inicial
Sequência de eventos	<ol style="list-style-type: none"> <li>1. O utilizador introduz o nome de utilizador e palavra-chave e autentica no Sistema de Relatórios e Análise de Tendências</li> <li>2. Validar a autenticação no lado do sistema</li> <li>3. O utilizador acede à página inicial do sistema</li> </ol>

Tabela 6. Caso de Uso - Estatística de nível por desporto de cada região

Identificador	USRT009_UC005
Nome	Estatística de nível por desporto de cada região
Descrição sumária	Visualização do relatório (gráfico e tabela cruzada) de nível dos desportistas por desporto numa região
Ator(es)	Utilizador
Pré-condições	O utilizador é autenticado no sistema
Pós-condições	Visualizar o relatório
Sequência de eventos	<ol style="list-style-type: none"> <li>1. O utilizador acede ao menu <i>Região</i></li> <li>2. O utilizador acede ao submenu <i>skill level dos desportistas por desporto</i></li> <li>3. O sistema abre a página de <i>Regiao_skill_level_desporto.aspx</i></li> <li>4. O utilizador seleciona um país</li> <li>5. O utilizador seleciona uma região</li> <li>6. O utilizador clica no botão <i>Gerar Relatório</i></li> <li>7. O sistema visualiza o relatório</li> </ol>

Tabela 7. Caso de Uso - Estatística da frequência dos eventos (por desporto) por dia em cada cidade

Identificador	USRT011_UC009
---------------	---------------

Nome	Estatística da frequência de evento (por desporto) por dia de cada cidade
Descrição sumária	Visualização do relatório (gráfico e tabela cruzada) de eventos por desporto mais ocorridos nos dias de semana numa cidade
Ator(es)	Utilizador
Pré-condições	O utilizador é autenticado no sistema
Pós-condições	Visualizar o relatório de frequência dos eventos
Sequência de eventos	<ol style="list-style-type: none"><li>1. O utilizador acede ao menu <i>Eventos</i></li><li>2. O utilizador acede ao submenu <i>Frequência dos eventos do desporto por dia</i></li><li>3. O sistema abre a página <code>Cidade_Frequencia_evento_desp_diadesemana.aspx</code></li><li>4. O utilizador seleciona um país</li><li>5. O utilizador seleciona uma região</li><li>6. O utilizador seleciona uma cidade</li><li>7. O utilizador seleciona data início dos eventos</li><li>8. O utilizador seleciona data fim dos eventos</li><li>9. O utilizador seleciona um desporto ou todos os desportos</li><li>10. O utilizador clica no botão <i>Gerar Relatório</i></li><li>11. O sistema visualiza o relatório</li></ol>

Os casos de uso dependem dos *user stories*; assim, quando houver alteração na parte de *user stories*, os casos de uso terão necessariamente que ser adaptados. Ambos, casos de uso e *user stories* normalmente podem sofrer alterações na fase de desenvolvimento, uma vez que é nesta fase que se identificam melhor as necessidades dos utilizadores.

## Resumo

Neste capítulo foi feita a especificação dos requisitos dos sistemas. Os requisitos permitem ter uma visão das funcionalidades que os sistemas vão oferecer, bem como quem serão os utilizadores que irão interagir com os sistemas a desenvolver.





# Capítulo 4

## Desenvolvimento

Neste capítulo descrevemos a arquitetura do sistema, o modelo de dados e a parte de implementação. A arquitetura do sistema será descrita a nível lógico e físico. Para o modelo de dados, será apresentado o esquema relacional. Relativamente à implementação descrevemos as abordagens utilizadas nas diferentes fases de desenvolvimento do Sistema de Recomendação e do Sistema de Relatórios e Análise de Tendências.

### 4.1. Arquitetura

Nesta subsecção apresentamos a arquitetura sob a qual desenvolvemos o sistema. Iremos descrever a arquitetura do Sistema de Recomendação (versão Demo) e do Sistema de Relatórios e Análise de Tendências. Para o Sistema de Recomendação, a arquitetura descrita é a da versão Demo, quando for integrado com o sistema existente (Playnify) será adaptada à arquitetura do sistema existente.

#### **Arquitetura Lógica**

A arquitetura lógica utilizada no Sistema de Recomendação e Sistema de Relatórios e Análise de Tendências é apresentada na Figura 4, e segue o modelo de camadas, ou seja, encontra-se dividida em camadas lógicas, onde cada camada disponibiliza um conjunto de serviços à camada superior. As camadas presentes na arquitetura lógica são a camada de apresentação, a camada de negócio e a camada de dados, para além dos componentes intermédios.

Na camada de apresentação encontra-se a interface do sistema, que é a camada com que o utilizador do sistema interage. Como se trata de um sistema web, esta camada é construída utilizando a tecnologia CSS3, HTML5 e AJAX.

A camada de negócio é a camada intermédia que coordena toda a aplicação, processa pedidos realizados pela camada de apresentação, realiza pedidos à camada de dados e devolve a página web à camada de apresentação.

Para o Sistema de Relatórios e Análise de Tendências, os dados pedidos pela camada de negócio através do componente NPGSQL são os dados utilizados para os filtros do relatório, ou seja, os dados do país, região, cidade e desporto. Da mesma maneira, para o Sistema de Recomendação (versão Demo), os dados de recomendações são pedidos através do componente NPGSQL. A camada de negócio apresenta o relatório através do componente CR Viewer. Para o Sistema de Recomendação e o Sistema de Relatórios e Análise de Tendências, os pedidos de dados através do NPGSQL são recuperados utilizando SQL de acordo com os eventos e procedimentos definidos com C# no *code behind*.

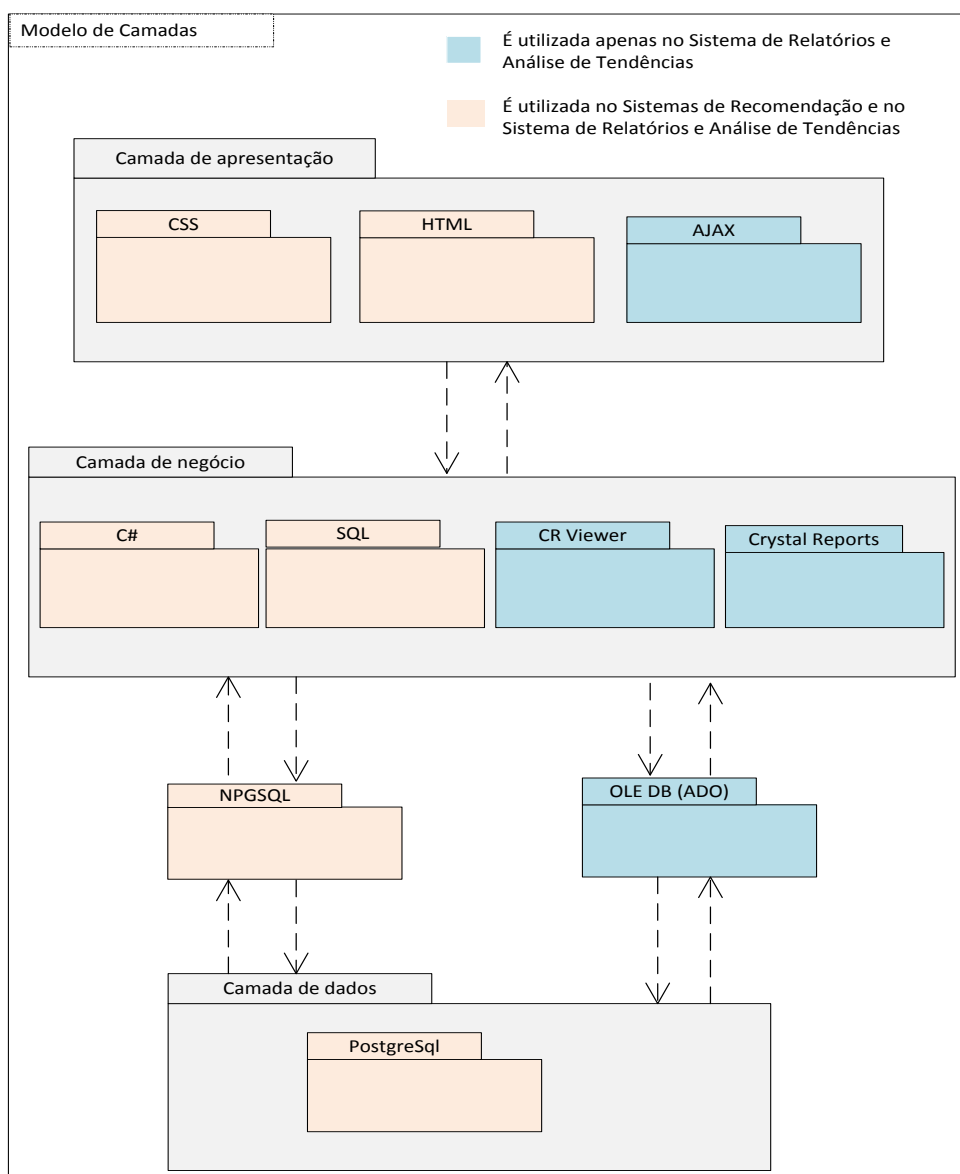


Figura 4. Arquitetura lógica

O componente Crystal Reports (CR) é o componente responsável pelo desenho de relatórios. Neste componente realiza-se o desenho de relatórios para o sistema de Relatórios e Análise de Tendências. Os relatórios são desenhados utilizando o Crystal Reports do SAP para Microsoft Visual Studio 2010. Os dados utilizados para relatórios são obtidos através do *Microsoft OLE DB provider for ODBC Drivers* do OLEDB (ADO) que comunica com a camada de dados.

A camada de dados é a camada da arquitetura que contém toda a informação dos sistemas. Nesta camada encontra-se o PostgreSQL que suporta a base de dados do sistema.

### Arquitetura Física

A arquitetura física descreve a camada física do sistema. Trata-se de uma arquitetura de alto nível do sistema de software, focando-se nas máquinas, conexões, componentes de software instalados nas máquinas e dependências entre componentes. A arquitetura física do sistema a desenvolver está ilustrada na Figura 5.

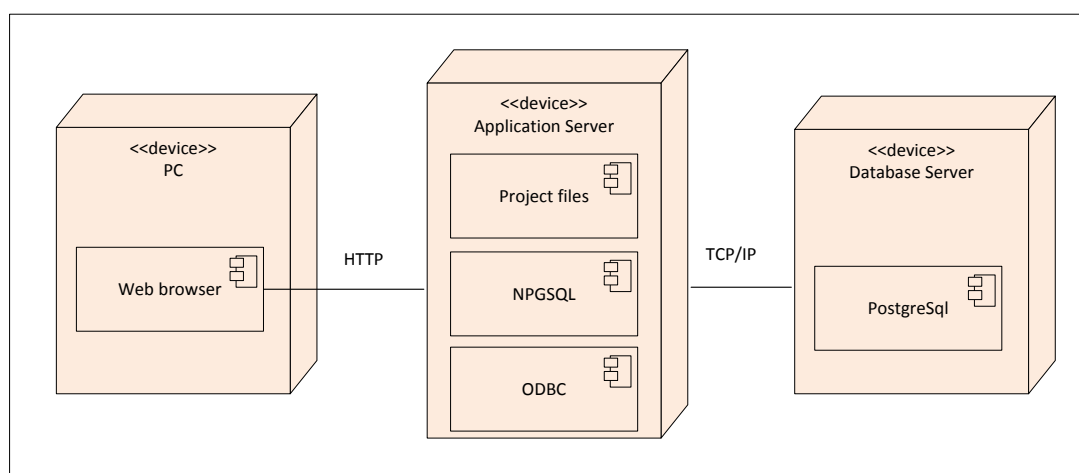


Figura 5. Arquitetura física

Os sistemas dividem-se em três nós, o dispositivo do cliente, o servidor da aplicação e a base de dados.

No dispositivo do cliente encontra-se o *web browser*, que é a única aplicação que o utilizador necessita para aceder ao serviço. A comunicação com o servidor da aplicação é realizado através do protocolo HTTP, utilizando a porta *standard*.

O servidor da nossa aplicação contém o IIS (*Internet Information Services*) como aplicação de servidor *web* para gerar páginas *web* do ASP.NET. Contém também os ficheiros referentes à aplicação, e para além disso, o DLL do NPGSQL e o ODBC, para ser possível comunicar com o servidor da base de dados.

O servidor de dados utiliza o PostgreSQL. A comunicação com o servidor da aplicação é realizada utilizando o protocolo TCP/IP (*Transmission Control Protocol / Internet Protocol*). O protocolo TCP/IP é referenciado na arquitetura física para mostrar que os nós necessitam de acesso à internet.

## Resumo

Os dois sistemas desenvolvidos no âmbito deste projeto de estágio seguem a arquitetura descrita em cima. Relativamente à arquitetura lógica há no entanto uma diferença no Sistema de Recomendação, onde os componentes “OLE DB (ADO)”, “AJAX” (da camada de apresentação) e “Crystal Reports e CR Viewer” (camada de negócio) não existem ( Na Figura 4 foi atribuída a cada conjunto de componentes uma cor específica).

## 4.2. Modelo de Dados

O Sistema de Recomendação e o Sistema de Relatórios e Análise de Tendências possuem uma base de dados comum. Nesta secção apresentamos o modelo de dados do sistema, recorrendo ao esquema relacional de bases de dados.

### Esquema Relacional

O esquema relacional tem como objetivo apresentar esquemas de relação, atributos, domínios, chave primárias, chave estrangeiras e outras regras de integridade. O Sistema de Recomendação e o Sistema de Relatórios e Análise de Tendências são desenvolvidos a partir da base de dados do Playnify (uma base de dados existente). Na Figura 6 apresentamos o esquema relacional apenas das tabelas utilizadas neste projeto. Para cada tabela apresentamos também apenas alguns atributos relevantes para o nosso projeto.

A tabela *auth\_user* contém os dados de acesso dos utilizadores (desportistas) do Playnify. A informação sobre os seus perfis é guardada na tabela *users\_userprofile*. A

tabela *users\_playerprofile* contém o perfil do utilizador como desportista, e a informação relativamente aos desportos que pratica é guardada na tabela *users\_playersportprofile*.

A localização geográfica dos utilizadores é guardada na tabela *geo\_location*. Esta tabela é uma tabela não normalizada (não cumpre a terceira forma normal), onde todas as informações sobre País, Região e Cidade são guardadas na mesma tabela.

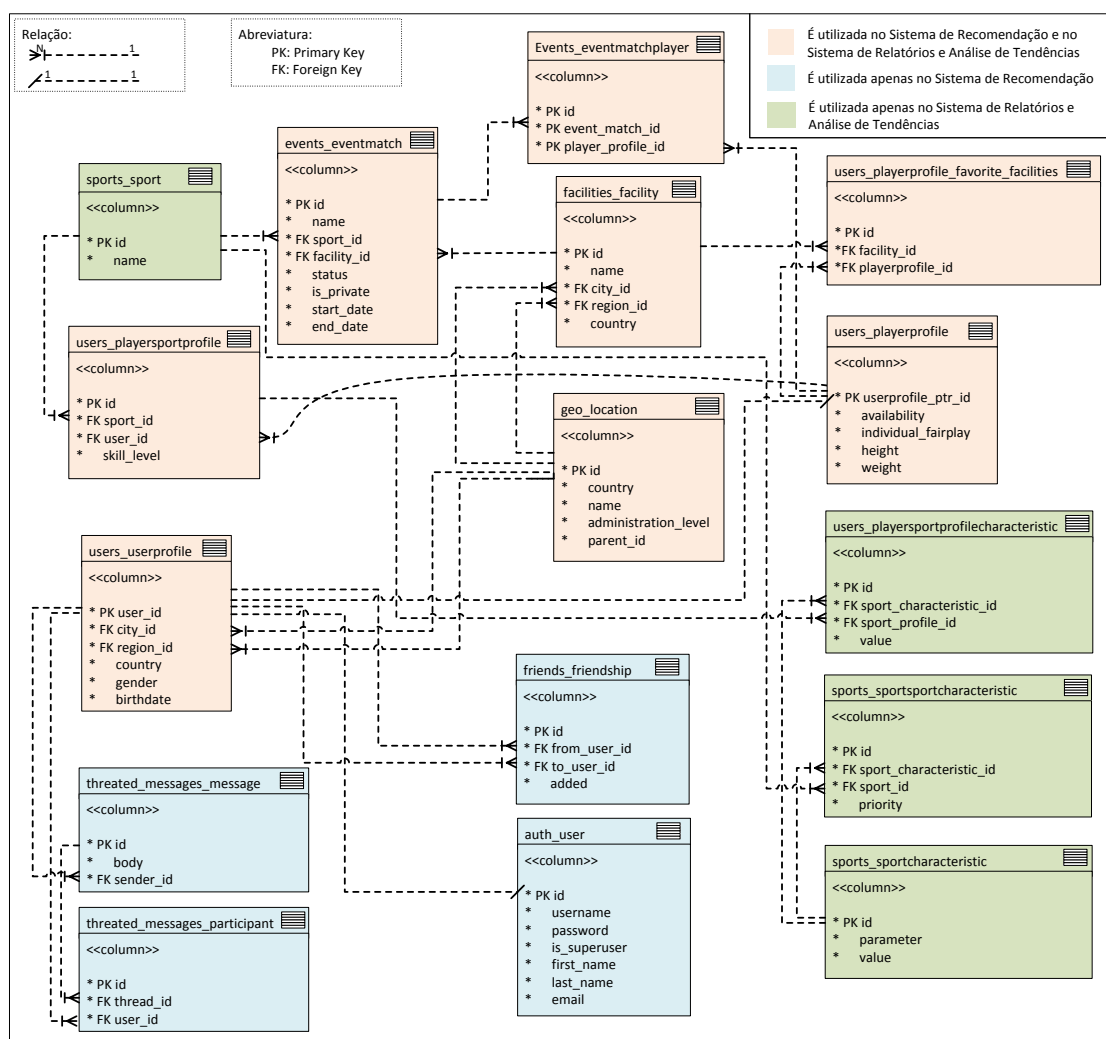


Figura 6. Modelo relacional

A tabela *sports\_sport* contém a informação sobre os desportos disponíveis no Playnify. As características disponíveis para descrever os diversos desportos são guardadas na tabela *sports\_sportcharacteristic*. A especificação das características de cada um dos desportos são guardadas na tabela *sports\_sportcharacteristic*. A informação sobre as características pessoais de um desportista, associadas a cada um dos desportos que pratica, são guardadas na tabela *users\_playersportprofilecharacteristic*.

A tabela *facilities\_facility* contém informação sobre instalações desportivas. A informação sobre as instalações que os desportistas marcaram como instalações favoritas é guardada na tabela *users\_playerprofile\_favorite\_facilities*.

A tabela *events\_eventmatch* contém informação sobre eventos. A informação sobre os desportistas que participam em cada evento é guardada na tabela *events\_eventmatchplayer*.

A tabela *friends\_friendship* contém a informação sobre as relações de amizade entre os desportistas. Os dados de amizade que estão nesta tabela são os dados dos desportistas que já são amigos. Os pedidos de amizade pendentes estão numa outra tabela semelhante.

A tabela *threatened\_messages\_message* contém informação sobre mensagens enviadas por desportistas dentro da plataforma. A informação sobre os utilizadores envolvidos é guardada na tabela *threatened\_messages\_participant*.

## Resumo

O modelo de dados representado na Figura 6 é um modelo geral para o Sistema de Recomendação e o Sistema de Relatórios e Análise de Tendências. Para diferenciar a sua utilização na parte de implementação, foi atribuída a cada conjunto de tabelas uma cor específica.

## 4.3. Modelo de Recomendação

Nesta subsecção apresentamos detalhadamente os algoritmos produzidos no desenvolvimento do protótipo versão Demo do Sistema de Recomendação.

### Recomendação de Desportistas

Um dos principais objetivos para um desportista se registar e utilizar o Playnify, é poder encontrar outros desportistas com quem praticar desporto juntos. Para isso, esses desportistas devem ter um certo perfil. Os desportistas a recomendar para um desportista alvo (utilizador ativo) são os amigos dos seus amigos que não são ainda seus amigos, aos quais ainda não enviou um pedido de amizade; moram na mesma cidade; praticam pelo menos um tipo de desporto comum ao mesmo nível; e têm disponibilidades que se intersectam marcadas na plataforma.

No Playnify, os desportistas relacionam-se uns com os outros através da amizade. A amizade é estabelecida através de pedidos enviados por um desportista a outros. Os pedidos de amizade dos desportistas estão ilustrados na Figura 7 e explicados a seguir.

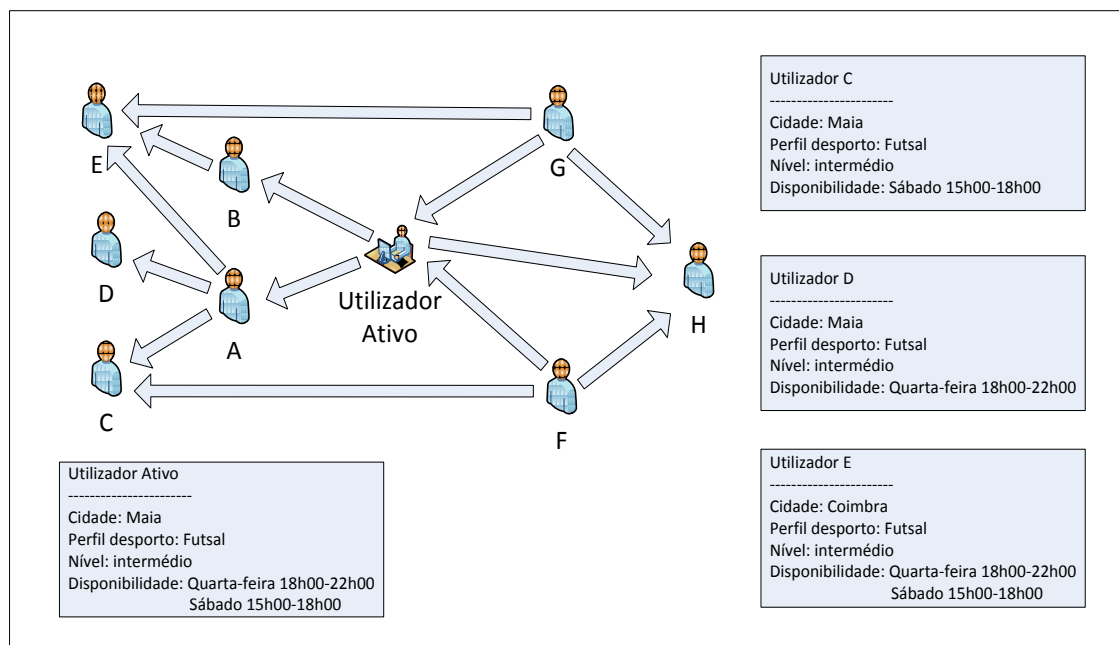


Figura 7. Amizade dos desportistas

A seta **Utilizador Ativo** → **A** significa que um pedido de amizade foi enviado pelo utilizador ativo (desportista alvo) e foi aceite pelo desportista A. Da mesma maneira, **G** → **Utilizador Ativo** significa que o pedido de amizade foi enviado por G e aceite pelo utilizador ativo.

De acordo com a informação sobre amizades dos desportistas registada na base de dados do Playnify e, para responder à preocupação dum desportista de encontrar os desportistas certos, desenvolvemos como solução o Sistema de Recomendação de “desportistas” que recomenda cada desportista alvo os (até) dez primeiros desportistas que satisfazem os critérios descritos anteriormente.

O algoritmo para a recomendação dos desportistas é ilustrado na Figura 8.

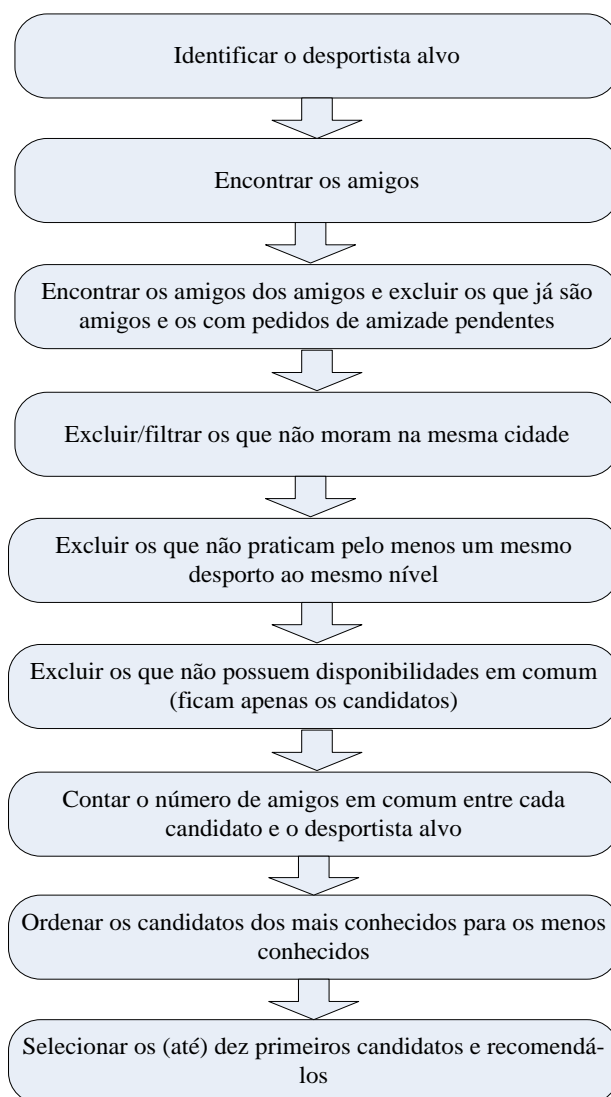


Figura 8. Diagrama do algoritmo de recomendação dos desportistas

Assim, quando aplicamos o algoritmo ao cenário de amizade na Figura 7, os desportistas que vão ser recomendados são o desportista C e o desportista D. O desportista E foi excluído devido à sua localização (ele vive numa cidade diferente do desportista alvo). Entre os desportistas C e D, como C tem mais amizades com amigos do utilizador alvo (C possui 2 amigos em comum e D possui apenas um), o desportista C vai aparecer primeiro na lista de recomendação. O código do algoritmo encontra-se no **Anexo A.1**.

### Recomendação de Eventos

Outro dos objetivos dos utilizadores do Playnify é encontrar eventos onde praticar desporto com outros desportistas registados. Para isso, o Playnify permite o registo da realização de eventos onde os utilizadores se podem inscrever.



Para ajudar os utilizadores do Playnify a encontrar eventos em que possam estar interessados, desenvolvemos o Sistema de Recomendação de “eventos”, que garante esta funcionalidade. Os eventos certos para um utilizador são os eventos que estão abertos para inscrição, que são públicos, que correspondem ao seu perfil de desporto e à sua disponibilidade, que estão a decorrer na sua cidade, e onde alguns desportistas já se inscreveram para participar.

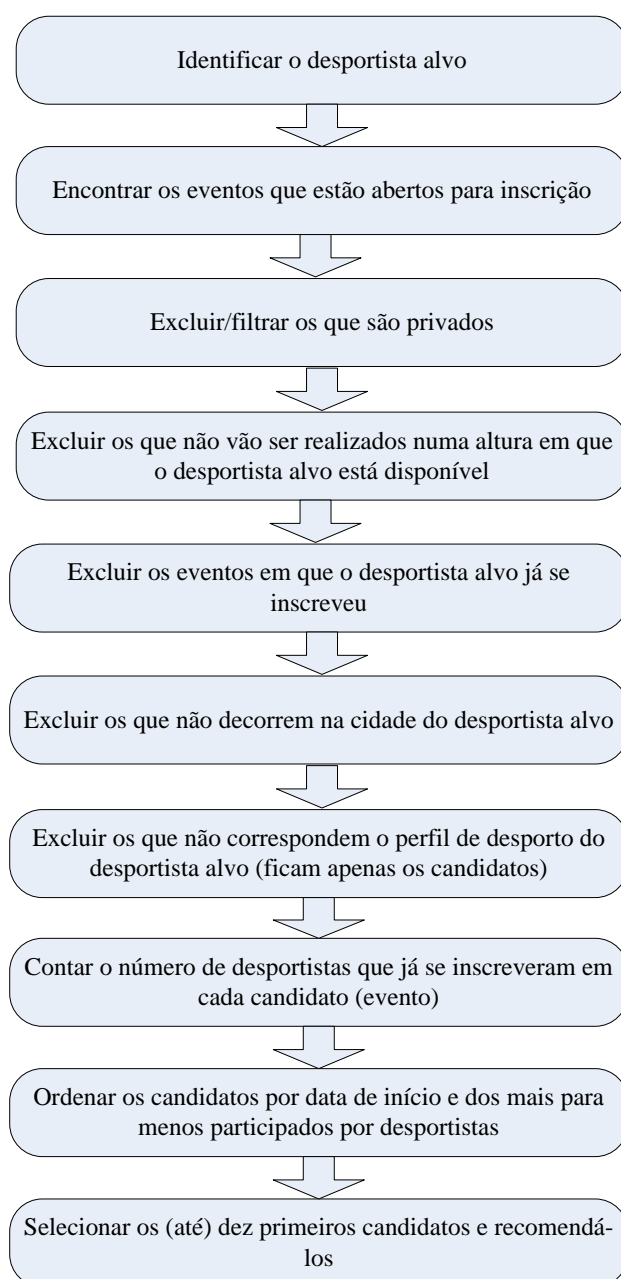


Figura 9. Diagrama do algoritmo de recomendação dos eventos

A solução proposta pelo sistema é procurar na base de dados do Playnify os eventos que satisfazem os critérios indicados em cima e recomendar os (até) dez primeiros

eventos por ordem crescente da data de início do evento e ordem decrescente do número de desportistas já inscritos.

O algoritmo de recomendação de eventos é ilustrado na Figura 9. É semelhante ao da recomendação dos desportistas, diferindo nos critérios e alguns passos de filtro dos candidatos.

### Recomendação de Instalações Desportivas

No Playnify, os eventos são realizados em instalações desportivas. Algumas instalações desportivas permitem realizar mais que um tipo de evento. Uma instalação desportiva

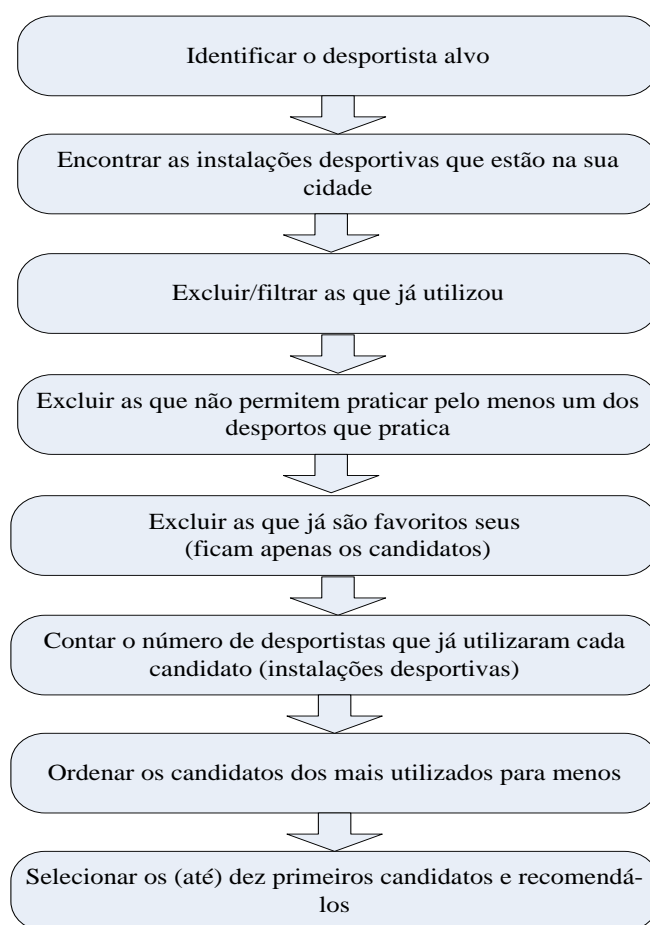


Figura 10. Diagrama do algoritmo de recomendação de instalação desportiva

certa para um desportista é uma instalação que está na sua cidade, que ainda não utilizou, que permite praticar o tipo de desporto que pratica, que ainda não foi marcada como favorita e que alguns desportistas já utilizaram.

Apresentamos na Figura 10 o funcionamento do algoritmo de recomendação da “instalação desportiva”.

## Implementação

Os algoritmos de recomendação são desenvolvidos em SQL, implementado no ASP.NET com *code behind* C#. O SQL recolhe as recomendações da base de dados através de eventos e rotinas desenvolvidos em C#, e visualizadas no *browser*, através de um protótipo desenvolvido no ASP.NET e formatado no CSS.

## Protótipo do Sistema de Recomendação da Versão Demo

As recomendações para cada desportista alvo são geradas de acordo com a sua identificação através da janela de autenticação do protótipo do Sistema de Recomendação, ilustrada na Figura 11.

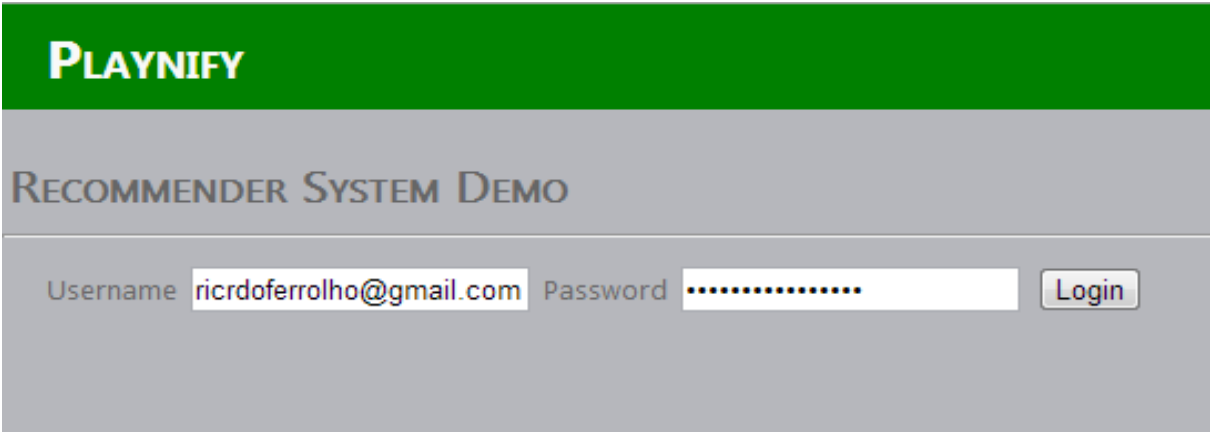


Figura 11. Página de autenticação do Sistema de Recomendação (versão Demo)

O desportista alvo é identificado através do seu email<sup>18</sup> introduzido. Baseado nesta identificação, as recomendações são geradas conforme os algoritmos descritos. As recomendações de “desportistas”, “eventos” e “instalações desportivas” geradas pelo sistema são exemplificadas na Figura 12. Note-se que, como informação adicional para o utilizador, as listas de recomendação fornecem ainda as identificações de cada item (ID) e o número de amigos e dos desportistas associados.

---

<sup>18</sup> O nome utilizador do Playnify é o email

PLAYNIFY									
RECOMMENDER SYSTEM DEMO									
Lists of Recommendations for <b>Ricardo Ferrolho</b>									
Recommended Players:			Recommended Events:				Recommended Venues:		
ID	Players	Friends Rate	ID	Events	Start Date	Players Rate	ID	Venues	Players Rate
5466	Flavio Da Silva Pereira	8	10	Futsal UW #001	24-10-2012 12:00:01	6	836	Campo futebol 7 esgueira	21
4994	Fábio Santos	7	38	Última Jornada de Futsal	29-10-2012 21:00:01	7	1000000022	Parque Gafanha da Nazaré	15
3701	Hugo Cura	7	55	Treino Equipa Futsal TSI	30-10-2012 19:30:01	15	1000000097	Campo urbanização Chave	12
2045	Diogo Gomes	7	84	Treino	03-11-2012 12:00:01	2	1000000106	Sede do BTT Brandoense	11
2222	António Costa	5	135	DQUA Champions League	19-11-2012 17:30:01	1	788	Clube Ténis Aveiro	9
4693	Jorge Farinha	3	144	UW 1 vs. UW 2	19-11-2012 19:00:01	3	1000000265	Pavilhão do CENAP	3
3291	Luis Paulo	3	314	Jogo Inicio do Ano	19-01-2013 22:00:01	3	623	Escola Ténis C Fitness	2
5426	Diogo Marto	2	545	Futsal UW #009	20-02-2013 12:00:01	12	1000000408	corrida cidade de Aveiro	1
5910	João Oliveira	1	595	Futsal UW#010	07-03-2013 12:00:01	10	634	C. Desp. S. Bernardo	1
			742	Futsal UW#011	15-03-2013 12:00:01	8	1000000023	ZoneFoot	1

Figura 12. Lista de recomendações

#### 4.4. Sistema de Relatórios e Análise de Tendências

Nesta secção apresentamos detalhes das abordagens utilizadas no desenvolvimento do Sistema de Relatórios e Análise de Tendências, algumas dificuldades encontradas e as soluções tomadas.

O Sistema de Relatórios e Análise de Tendências desenvolvido neste projeto de estágio é composto por duas partes: Estatísticas de Relatórios e Estatísticas de Análise de Tendências. Estatísticas de Relatórios são desenvolvidas sobretudo, para monitorizar estatisticamente os utilizadores, desportos, eventos, marcas dos equipamentos e instalações desportivas do Playnify. Estatísticas de Análise de Tendências são desenvolvidas para identificar as tendências e os hábitos dos utilizadores do Playnify.

Na nossa solução, uma página do Sistema de Relatórios e Análise de Tendências é composta por filtros e relatórios. Um filtro é o conjunto dos critérios que determinam a informação visualizada no relatório. Um relatório é a estatística visualizada de acordo com critérios definidos nos filtros. O filtro é desenvolvido no ASP.NET com *code behind* C#, e o relatório é desenvolvido no Crystal Reports (CR).

##### Filtros

A primeira parte do relatório são os filtros. Os filtros são utilizados para filtrar a informação visualizada no relatório. Os filtros do sistema são país, região, cidade, desporto e período de tempo.

Na nossa solução, como os filtros aparecem em todos os relatórios, desenvolvemos três mecanismos de controle (*user controls*<sup>19</sup>), que são a base dos filtros para os relatórios do sistema. Estes *user controls* são desenvolvidos para facilitar a reutilização do mesmo filtro em vários relatórios. Os três *user controls* são:

1) *Location Control*: este *user control* é composto por filtros de “país”, “região” e “cidade”. Como todos os relatórios são filtrados ou “por país”, ou “por região”, ou “por cidade”, foi introduzido um parâmetro neste mecanismo de controle que determina que itens (país, país e região, ou país, região e cidade) vão ser mostrados.

O parâmetro tem valores possíveis (níveis 1, 2 ou 3) representando respetivamente as opções “país” (nível 1), “região” (nível 2) e “cidade” (nível 3). Para cada nível desenvolvemos uma função que recolhe a informação da base de dados e outra função que devolve o ID do elemento da lista que foi selecionado pelo utilizador. O ID devolvido é que é depois utilizado para filtrar a informação visualizada no relatório. O código do *user control* da localização geográfica pode ser consultado no **Anexo A.2**.

A lista para cada filtro é obtida da base de dados através do NPGSQL usando SQL e, invocado no *code behind* C#. A atualização da lista do filtro é feito através do AJAX que faz uma atualização direta.

Para utilizar este *user control* num relatório, foi necessário incluir no topo da página ASPX uma linha de código como exemplificado a seguir:

```
<%@ Register Src="~/LocationControl.ascx" TagName="Cteid" TagPrefix="C" %>
```

Aqui, *LocationControl.ascx* é o nome do ficheiro do *user control*. *TagName* e *TagPrefix* são respetivamente o nome e prefixo do *Tag* atribuídos à referência.

Com isso é possível referenciar o *user control* no corpo da página através do *Tag* para ser visualizado. Neste caso, utiliza-se código como o seguinte:

```
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
  <div>
    <C:Cteid ID="CCTEID" runat="server"/>
  </div>
</asp:Content>
```

---

<sup>19</sup> Estes mecanismos de controle foram criados da mesma forma de uma página web no ASP.NET.

Finalmente, utiliza-se o ID definido nesta página no *code behind C#* para definir o nível do filtro que pretendemos aplicar. No código em baixo mostramos um exemplo de nível 3 (nível “cidade”) para a página acima que mostra o filtro na página enquanto a página é carregada.

```
protected void Page_Load(object sender, EventArgs e)
{
    CCTEID.Parametro(3);
}
```

*CCTEID* é o ID atribuído ao filtro no corpo da página e *Parametro(3)* é a função que chama função *Parametro* do nível 3 definido em *LocationControl.ascx*.

A lista do filtro da localização (lista do país) é ilustrada na Figura 13.



Figura 13. Lista do filtro de país

2) *Date Control*: é um *user control* para as datas de início e as datas finais dos eventos. Já que cada evento é realizado dentro de um determinado período de tempo. Este *user control* é desenvolvido para poder ser utilizado em vários relatórios. A sua utilização na página dum relatório é feita da mesma maneira como a do *user control* da localização geográfica. A janela do calendário utilizada nos filtros de data é ilustrada na Figura 14.

Abril						
Maio de 2013						
Junho						
seg	ter	qua	qui	sex	sáb	dom
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Figura 14. Calendário utilizado no filtro de data

3). *Sport Control*: é um *user control* para a seleção de um desporto. Este *user control* é desenvolvido para o filtro de desporto, utilizado em vários relatórios. É composto por uma função para recolher os dados de desporto na base de dados e outra função para devolver o ID do elemento da lista selecionado pelo utilizador. A sua utilização na página dum relatório é idêntica à utilização dos *user controls* anteriores. A lista de filtro do desporto é ilustrada na Figura 15.

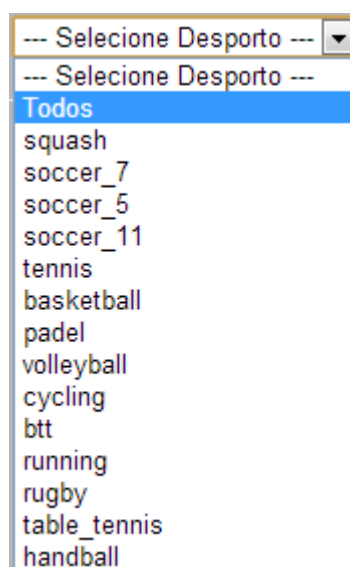


Figura 15. Lista de filtro do desporto

Um dos elementos da lista que não faz parte da base de dados é o elemento “*Todos*”. Inserimos este elemento para permitir incluir todos os desportos no relatório quando o utilizador escolher esta opção.

O filtro do desporto, da data e da localização (nível 3) visualizado na página é apresentado na Figura 16.

Figura 16. Filtros do Sistema de Relatórios e Análise de Tendências

Note-se que, atribuímos valores por omissão ao filtro de localização (país="Portugal", região="Porto" e cidade="Porto"), filtro de data (data de início e data fim = data de hoje) e filtro de desporto (desporto="todos") para, quando o utilizador não os definir retornar esses valores. Para o filtro de data, o sistema avisa quando o utilizador escolhe uma data fim menor do que a data de início.

## Desenho de Relatórios

Para desenhar um relatório, o CR tem de comunicar com a base de dados. Assim, configuramos primeiro o *driver* de ODBC. A configuração é feita no DNS do ODBC que é composto por *nome do servidor*, *nome da base de dados*, *número da porta*, *utilizador* e *palavra-chave da base de dados*.

Para o desenho dos relatórios no CR começou-se por escolher o nome da fonte de dados atribuído na configuração no ODBC. Escolheram-se as tabelas a seguir utilizadas no relatório e as relações entre elas no CR. Depois, desenvolveu-se o relatório com base nestas tabelas. Um exemplo da relação entre as tabelas dum relatório desenvolvido no CR está no **Anexo A.3**.

Além de desenhar os relatórios, definiu-se também no CR os parâmetros utilizados para comunicar com os filtros desenvolvidos no ASP.NET e utilizaram-se estes parâmetros para filtrar a informação apresentada nos relatórios. Também foram definidas as fórmulas que se aplicam na transformação de dados brutos em informação útil e que interpretam os filtros enviados por ASP.NET. Uma das fórmulas que transforma dados brutos em informação útil é apresentada em baixo.

```
if DayOfWeek({events_eventmatch.start_date})=1 then "1-Segunda-Feira"
else if DayOfWeek({events_eventmatch.start_date})=2 then "2-Terça-Feira"
else if DayOfWeek({events_eventmatch.start_date})=3 then "3-Quarta-Feira"
else if DayOfWeek({events_eventmatch.start_date})=4 then "4-Quinta-Feira"
else if DayOfWeek({events_eventmatch.start_date})=5 then "5-Sexta-Feira"
else if DayOfWeek({events_eventmatch.start_date})=6 then "6-Sábado"
else if DayOfWeek({events_eventmatch.start_date})=7 then "7-Domingo"
```

Esta fórmula, escrita com sintaxe de CR, associa a cada data e hora (vindo da base de dados) o dia da semana correspondente.

## Integração do CR com ASP.NET

A integração do relatório desenvolvido no CR com a página web do ASP.NET é feita através do CR Viewer, que é o componente .NET do CR. Quando se insere o



componente CR Viewer na página de ASPX, automaticamente a referência ao componente CR é registada no topo da página com o prefixo de *tag* CR.

```
<%@ Register assembly="CrystalDecisions.Web, Version=13.0.2000.0, Culture=neutral,
PublicKeyToken=692fbea5521e1304" namespace="CrystalDecisions.Web" tagprefix="CR"
%>
```

Para invocar o relatório, é necessário definir o nome do relatório (da extensão **rpt**) desenvolvido com CR dentro do componente *report* do *CrystalReportSource*. Um exemplo de invocar o relatório *Cidade\_Frequencia\_evento\_por\_desp\_diadesemana.rpt* é o seguinte:

```
<CR:CrystalReportSource ID="CrystalReportSource1" runat="server">
  <Report FileName="Cidade_Frequencia_evento_por_desp_diadesemana.rpt">
  </Report>
</CR:CrystalReportSource>
```

Para visualizar o relatório no CR Viewer, é preciso chamar o ID atribuído ao relatório no *CrystalReportSource* dentro do componente *CrystalReportViewer*.

```
<CR:CrystalReportViewer ID="CidadeFrequenciaEventoDia" runat="server"
ReportSourceID="CrystalReportSource1"
AutoDataBind="True" Height="1202px" Width="1104px"
PrintMode="ActiveX" />
```

## Associar os parâmetros dos Filtros com os do Relatório

Para os filtros poderem ser aplicados num relatório, temos de definir duas partes:

1) No *code behind* C#: nesta parte associamos os valores retornados dos elementos seleccionados nos filtros com os parâmetros definidos no relatório.

```
// Instanciar o relatório
Cidade_Frequencia_evento_por_desp_diadesemana C_freq_eve_Desp_dia = new
Cidade_Frequencia_evento_por_desp_diadesemana();

// Associar os parâmetros do filtro com o relatório
C_freq_eve_Desp_dia.SetParameterValue("Cidade", CFEDD.cidade());
```

2) Na fórmula da seleção de registo no relatório: nesta parte definimos as fórmulas que interpretam os parâmetros recebidos dos filtros para serem interpretados e aplicados no filtro da informação visualizada no relatório.

```
{geo_location.id} = ToNumber({?Cidade})
```

O valor retornado do elemento seleccionado na lista do filtro é definido com o tipo de dados *string*. Neste caso, convertemo-lo, na fórmula de seleção de registo no relatório,

para o tipo de dados numérico inteiro, pois o ID de cidade na base de dados tem tipo inteiro.

### Integração de Filtros e Relatório na Página ASPX do Sistema

Recapitulamos que, o relatório é desenvolvido em CR e os dados utilizados para desenvolver os relatórios são obtidos através do ODBC; os filtros são desenvolvidos com *user controls* do ASP.NET e os dados utilizados nos filtros são recuperados utilizando SQL através de *data provider* NPGSQL; e que a página *web* do sistema é também desenvolvida com ASP.NET(*code behind* C#). Assim, para o Sistema de Relatórios e Análise de Tendências ficar a funcionar, cada relatório tem de integrar com os filtros numa página *web*.

A integração dum relatório com os filtros da página do sistema segue os seguintes passos (consultar o código do relatório no **Anexo A.4**):

- 1) Inserir na página ASPX do sistema os filtros desenvolvidos nos *users control*;
- 2) Inserir na página ASPX do sistema o relatório desenvolvido no CR através do CR Viewer;
- 3) Associar os filtros com os parâmetros do relatório no *code behind* C#;
- 4) Definir as fórmulas da seleção de registo no relatório desenvolvido com CR para interpretar no relatório os códigos dos filtros recebidos.

### Relatórios do Sistema de Relatórios e Análise de Tendências

Os relatórios do Sistema são compostos por estatísticas de Relatórios e estatísticas de Análise de Tendências. Um relatório do Sistema de Relatórios e Análise de Tendências é geralmente composto por um gráfico e uma tabela cruzada.

As estatísticas de Relatórios apresentam informação sobre quais os desportos mais praticados num país, quais o *skill level* dos desportistas por desporto numa região, quantos desportistas por desporto numa cidade categorizado por sexo, quais as marcas dos equipamentos mais utilizadas num desporto, etc.

Ilustramos na Figura 17 uma estatística de Relatórios de *skill level* dos desportistas por desporto; no **Anexo A.5** uma estatística dos desportistas por desporto; no **Anexo A.6** uma estatística de instalações desportivas favoritas.

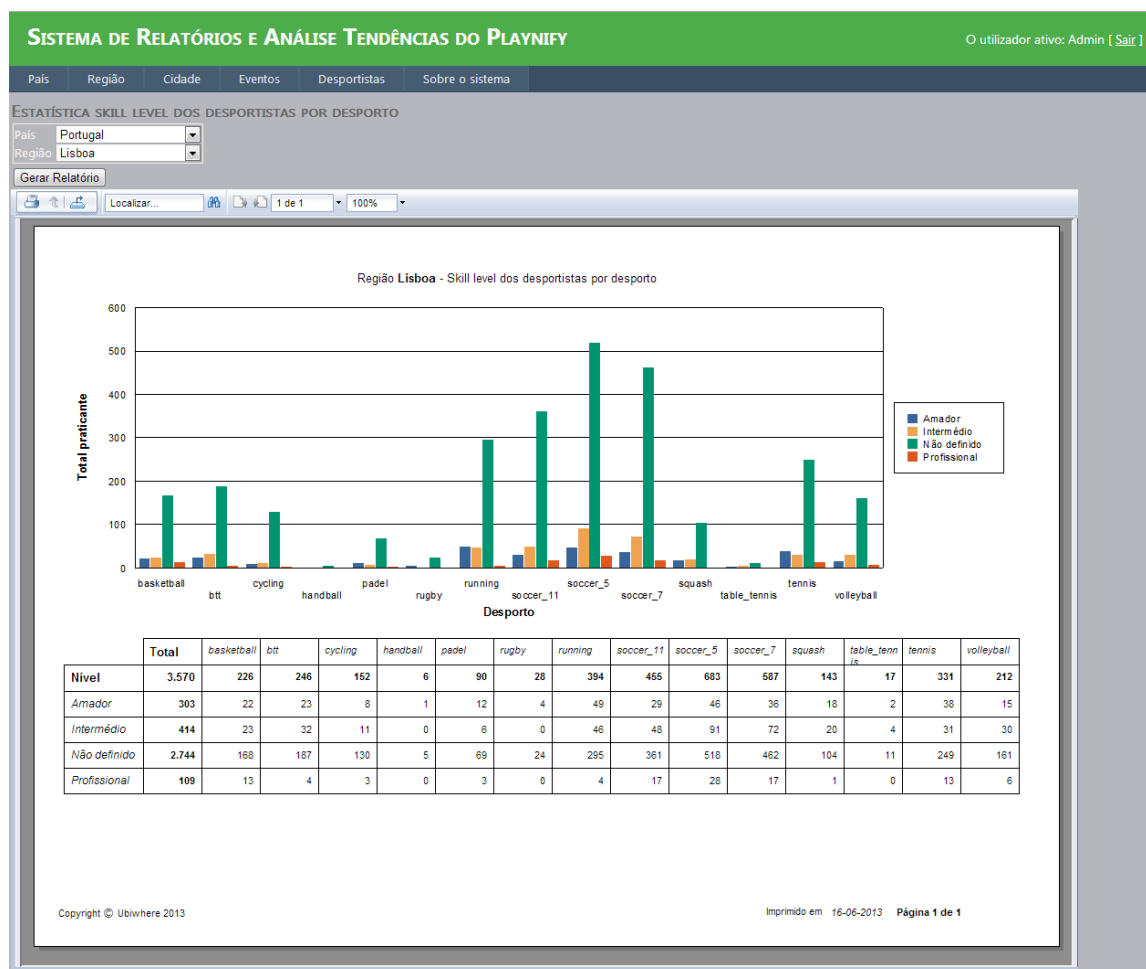


Figura 17. Estatística de Relatórios de skill level dos desportistas por desporto

As estatísticas da Análise de Tendências apresentam informação sobre as tendências de praticar desportos dentro dum período de tempo. Por exemplo: para saber em que dia da semana é que os desportistas praticam mais futebol em Portugal no verão; ou qual é o desporto mais praticado na região Porto no verão e em que hora; etc.

Ilustramos na Figura 18 uma estatística de Análise de Tendências categorizada por “dia”, no **Anexo A.7** uma estatística categorizada por “mês”, no **Anexo A.8** uma estatística categorizada por “semana”, e no **Anexo A.9** uma estatística de Análise de Tendências categorizada por “hora”.

Todos os relatórios permitem ser imprimidos e exportados para RPT, PDF, Microsoft Word, Microsoft Excel, RTF, CSV e XML.

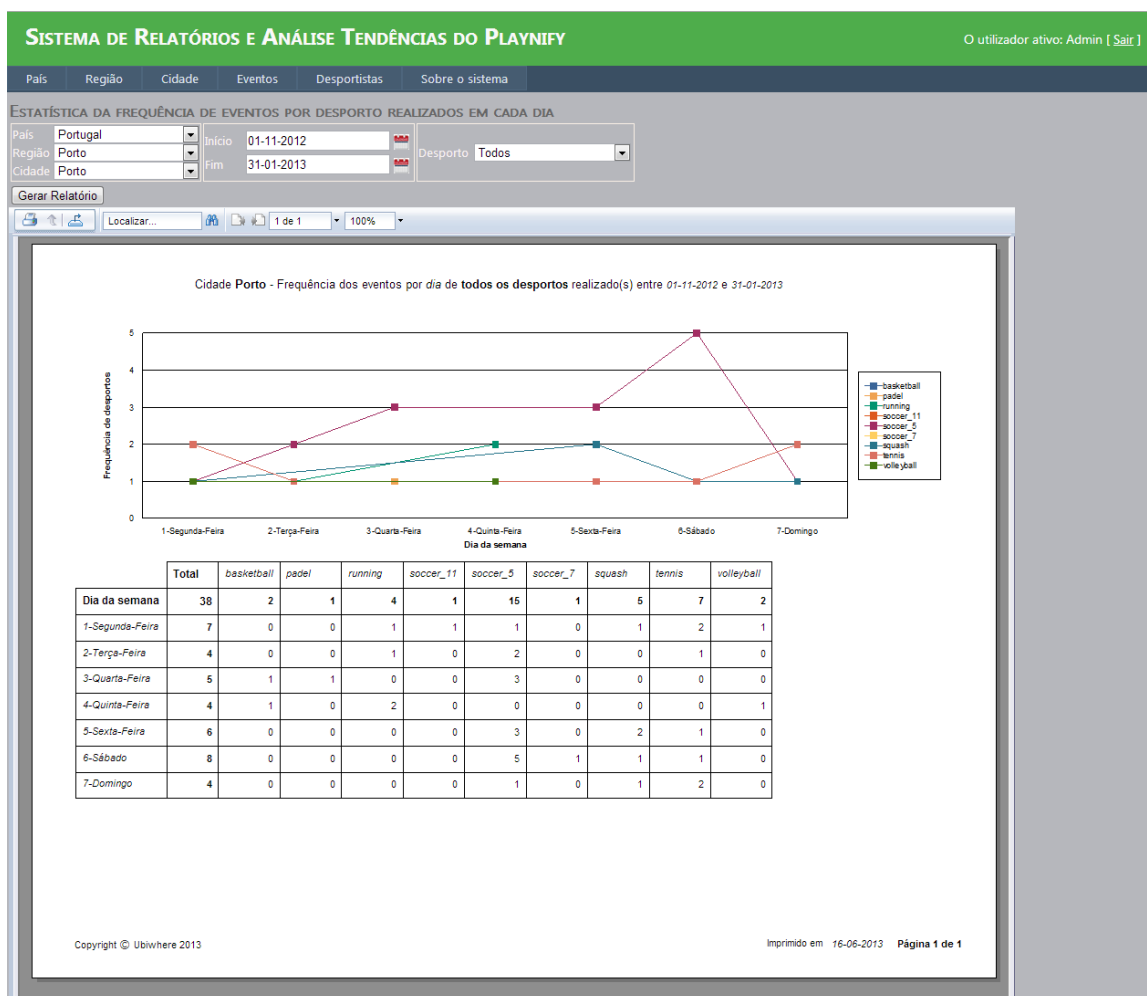


Figura 18. Estatística de Análise de Tendências categorizada por dia

## Módulos principais do Sistema de Relatórios e Análise de Tendências

O Sistema de Relatórios e Análise de Tendências é composto por uma página de autenticação, uma página principal (*master page*<sup>20</sup>) e a página do relatório.

A página de autenticação é utilizada para validar o acesso dos utilizadores. Neste sistema, apenas os administradores do Playnify têm permissão de acesso. A página de autenticação é ilustrada na Figura 19.

<sup>20</sup> Uma página que permite-nos criar um *layout* consistente para a página da aplicação.

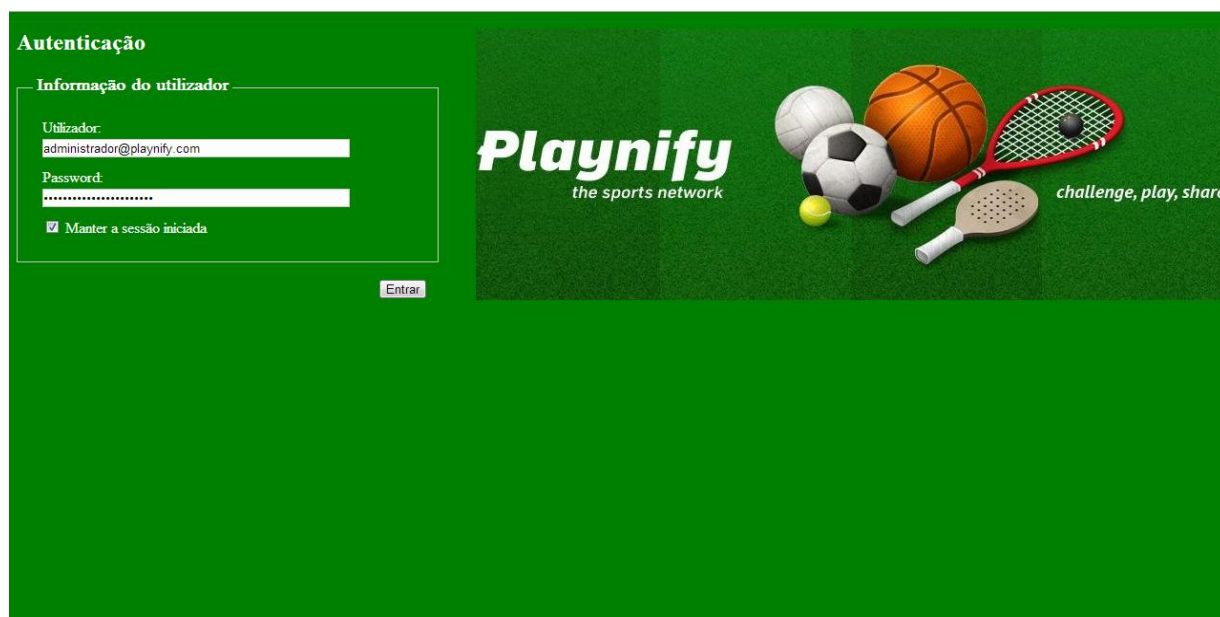


Figura 19. Página de autenticação do Sistema de Relatórios e Análise de Tendências

A página principal é uma página com um cabeçalho composto pelo nome do sistema e a informação do utilizador ativo, os menus e submenus, e o corpo da página para visualizar o relatório do sistema/a página do relatório (Figura 20). A ilustração detalhada pode ser consultada no **Anexo A.10**.

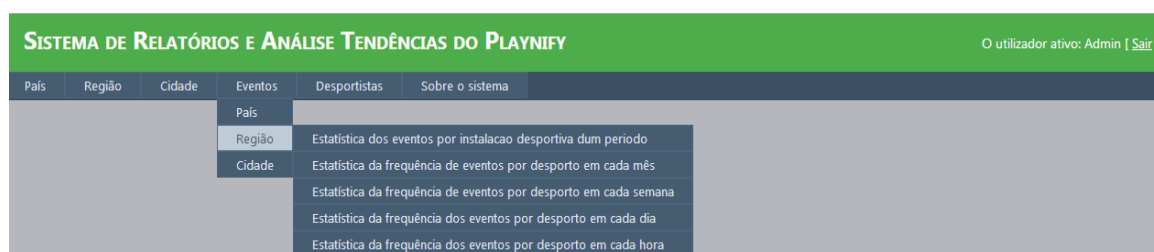


Figura 20. Página principal, menus e submenus do Sistema de Relatórios e Análise de Tendências

A Figura 20 mostra a estrutura do relatório do Sistema de Relatórios e Análise de Tendências organizada em menus e submenus.

### Estrutura de Relatórios do Sistema de Relatórios e Análise de Tendências

O relatório do sistema é composto por um gráfico, por uma tabela cruzada, ou ambos. Todos os relatórios podem ser filtrados por “País”, “Região” e/ou “Cidade”. Adicionalmente existem filtros para “Desporto” e “Período de tempo”.

## Dificuldades Principais e Soluções Adoptadas

Encontramos várias dificuldades durante o desenvolvimento do projeto. Uma dificuldade principal foi a definição de campos de filtro da localização geográfica no Crystal Reports (CR). Como a tabela de localização geográfica (*geo\_location*) não é uma tabela normalizada (não cumpre a terceira forma normal), onde os registos do “país”, “região” e “cidade” são registados na mesma tabela, é difícil fazer um filtro integrado com o CR, utilizando campos de parâmetro do CR para serem aplicados no filtro de seleção de registo do relatório. De facto, existem dependências entre os atributos “país”, “região” e “cidade”, e numa base de dados relacional estes registos deveriam estar registados em três tabelas diferentes. Por outro lado, o CR, também não permite ter mais do que uma ligação/relação entre duas tabelas o que não nos permite utilizar os atributos país, região e cidade como filtros integrados dum relatório no CR.

A base de dados (BD) do Playnify é uma BD existente que não nos permite fazer alterações a nível da arquitetura da BD. Para contornar este problema, desenvolvemos os filtros através do *user controls*. Separamos os registos do país, região e cidade na tabela *geo\_location* utilizando os *queries* do SQL. Começamos por obter os registos de “país”, utilizamos o ID de cada país selecionado pelo utilizador para obter os registos de “região” correspondentes, e utilizamos o ID da região selecionada pelo utilizador para obter os registos de “cidade” associados. Desta forma, conseguimos resolver o problema de integração dos filtros com os relatórios. Assim, desenvolvemos os filtros e os relatórios separadamente e desenvolvemos as rotinas para os integrar.

## Deployment

Para o sistema poder funcionar como cliente-servidor, é necessário primeiro instalar e configurar o servidor da aplicação. Para isso, instalámos e configurámos o IIS (*Internet Information Services*), o Microsoft .NET Framework 4, o *psqlODBC* (o ODBC driver do PostgreSQL) e o *runtime* do Crystal Reports (*SAP Crystal Reports Runtime Engine for .NET Framework*). O *psqlODBC* e *runtime* do Crystal Reports são os requisitos principais para o Sistema de Relatórios e Análise de Tendências poder funcionar no servidor.

Relativamente ao *deployment* do Sistema de Recomendação versão Demo e do Sistema de Relatórios e Análise de Tendências, foram gerados os pacotes das aplicações (*deployment packages*) e lançados no servidor. O ficheiro de configuração do Sistema Relatórios e Análise de Tendências está no **Anexo A.11**.

# Capítulo 5

## Testes e Avaliação

Neste capítulo descrevemos os testes realizados para verificar o bom funcionamento do Sistema de Recomendação e do Sistema de Relatórios e Análise de Tendências. Foi também feita uma avaliação dos dois sistemas descrita na Seccão 5.2.

### 5.1. Testes

Os testes foram feitos para avaliar as funcionalidades do Sistema de Recomendação e do Sistema de Relatórios e Análise de Tendências, descobrir e corrigir *bugs* e garantir que as recomendações geradas e informações visualizadas sejam de acordo com os conceitos descritos nos capítulos anteriores. Os testes para ambos, o Sistema de Recomendação e o Sistema de Relatórios e Análise de Tendências, foram feitos com base em casos de teste. Adoptamos o modelo dos casos de teste que a empresa Ubiwhere utiliza para testar os seus produtos. Um caso de teste é composto por descrições do objetivo de teste, pré-condição (condição antes da execução do teste), passos de teste e pós-condição.

#### **Sistema de Recomendação**

Para os testes do Sistema de Recomendação foram escolhidos três utilizadores ativos (desportistas alvo) com diferentes localizações geográficas (cidades), modalidades de desporto e disponibilidades para jogar já marcadas na plataforma. Em seguida, foram escolhidos seis elementos (dois elementos de cada categoria desportistas/eventos/instalações deportivas) para correr os testes. Utilizámos os dois elementos de cada categoria para testar com todos os três utilizadores ativos, verificando se foram ou não corretamente recomendados. Exemplificamos alguns casos de teste representativos na Tabela 8.

Tabela 8. Casos de teste representativos do Sistema de Recomendação.

ID	Objetivo	Pré-condição	Passos de teste		Pós-condição
			Descrição	Resultado esperado	
SRCT1	Autentificar um utilizador ativo	O utilizador não está autenticado no Sistema de Recomendação	1. O utilizador abre a página de autenticação do Sistema de Recomendação versão Demo	O sistema exibe a janela de autenticação do Sistema de Recomendação	O utilizador é autenticado com sucesso
			2. O utilizador introduz o nome do utilizador <i>ricrdoferrolho@gmail.com</i> e clica no botão <i>Login</i>	O sistema exibe a página principal do Sistema de Recomendação	
SRCT2	Recomendar ao utilizador ativo o desportista com ID=5466	O utilizador já está autenticado no Sistema de Recomendação	1. Confirma o nome do utilizador ativo	O nome do utilizador ativo é <i>Ricardo Ferrolho</i>	O desportista com ID=5466 é um dos desportistas recomendados ao utilizador ativo
			2. Confirma na lista de <i>recommended players</i> a existência do desportista com ID=5466	O desportista com ID=5466 é encontrado na lista de <i>recommended players</i>	

Depois de corrigidos alguns *bugs* encontrados, todos os casos de teste realizados correram com sucesso, tendo os algoritmos do Sistema de Recomendação funcionado de acordo com os critérios especificados no Capítulo 3.



## Sistema de Relatórios e Análise de Tendências

Para as funcionalidades do Sistema de Relatórios e Análise de Tendências percorremos todos os menus e submenus do sistema e confirmámos a correção do funcionamento dos filtros e da informação visualizada nos relatórios que o sistema fornece. Na Tabela 9 apresentamos alguns casos de teste representativos.

Tabela 9. Casos de teste representativos do Sistema de Relatórios e Análise de Tendências

ID	Objetivo	Pré- condição	Passos de teste		Pós- condição
			Descrição	Resultado esperado	
SRTCT1	Autenticar utilizador	O utilizador não está autenticado no Sistema de Relatórios e Análise de Tendências	1. O utilizador abre a página de autenticação do Sistema de Relatórios e Análise de Tendências	Sistema exibe a janela de autenticação do Sistema de Relatórios e Análise de Tendências	O utilizador é autenticado com sucesso
			2. O utilizador introduz o nome do utilizador <i>administrador@playnify.com</i> e a palavra-chave e clica no botão <i>Entrar</i>	O sistema exibe a página principal do Sistema de Relatórios e Análise de Tendências	
SRTCT2	Visualizar a estatística <i>total dos desportistas por desporto</i> em Portugal	O utilizador está autenticado no Sistema de Relatórios e Análise de Tendências	1. O utilizador vai ao menu <b>País</b> , clica no submenu <i>Estatística dos desportistas por desporto</i>	O sistema exibe na página principal ( <i>Master page</i> ) uma página com título <i>Estatística total dos desportistas por desporto</i> , o filtro de “País” com o valor por omissão “---	A estatística é visualizada com sucesso

				<b>Selecione País---</b> “ e o botão <i>Gerar Relatório</i>	
			2. O utilizador escolhe <b>Portugal</b> e clica no botão <i>Gerar Relatório</i>	O sistema visualiza o relatório através de um gráfico de barra de cada desporto para todos os desportos e uma tabela cruzada contendo os números totais e respetivas percentagens correspondente à informação apresentada no gráfico.	

Encontrámos também alguns *bugs* durante os primeiros testes do Sistema de Relatórios e Análise de Tendências. Estes foram corrigidos e foi verificado que agora todos os casos de teste executam com sucesso.

## 5.2. Avaliação

Nesta secção descrevemos as técnicas de avaliação das recomendações para o Sistema de Recomendação e das informações visualizadas no relatório para o Sistema de Relatórios e Análise de Tendências.

Para a avaliação do Sistema de Recomendação, validámos as listas de recomendação e avaliámos as recomendações geradas. As listas de recomendação de cada categoria foram validados utilizando SQL para verificar todas as condições aos quais o sistema deve obedecer. Escolhemos alguns utilizadores para realizar a validação. Como

exemplo, apresentamos no **Anexo A.12** os *queries* utilizados para validar a correção de recomendação do desportista com ID=5466, que o sistema recomendou ao utilizador *Ricardo Ferrolho* (ID=2022).

Na avaliação das recomendações geradas prevíamos usar uma técnica de avaliação *offline* experimental que funciona por dividir os dados atuais em duas partes: passado e futuro. Os dados de passado são usados para treino dos algoritmos, ou seja, para gerar recomendações e analisá-los por comparação com os dados de futuro. Contudo, como este Sistema de Recomendação se baseia em localização, perfil de utilizadores e dados históricos de amizade, descobrimos através dos testes que temos um problema de “*dispersão*”. Dispersão é um dos desafios dos algoritmos *User-Based Collaborative Filtering*, descrito em [22], onde um sistema de recomendação pode não recomendar nenhum item para um utilizador específico. Isto, porque até ao momento o número de utilizadores registados do Playnify não é suficientemente grande (menos do que 7000 utilizadores dispersos por locais diferentes). Tratando-se no caso do nosso Sistema de Recomendação de uma versão Demo, a avaliação do algoritmo e do seu desempenho será feita quando for integrado com o Playnify.

Para analisar o Sistema de Relatórios e Análise de Tendências foram escolhidas várias entradas nos filtros e depois verificada a correção da informação nos relatórios obtidos. A validação da informação visualizada nos relatórios foi também feita através de *queries* SQL. No **Anexo A.13** podem ser consultados os *queries* que utilizamos para validar a *estatística dos desportistas por desporto em Portugal*. Estas operações foram aplicadas para todos os relatórios do sistema. Relativamente à compatibilidade do sistema com diferentes *browsers*, validámos a compatibilidade com o Internet Explorer, o Chrome e o Mozilla Firefox. Os resultados obtidos são satisfatórios, visto que todas as funcionalidades testadas se comportam como esperado.

Os testes e as validações foram efetuados não só pelo autor, mas também pela restante equipa da Ubiwhere, existindo ainda registo formal dos mesmos.



# Capítulo 6

## Conclusão e Trabalho Futuro

Neste capítulo descrevemos algumas conclusões, assim como algumas linhas possíveis para trabalho futuro que consideramos importante.

### 6.1. Conclusão

O objetivo principal deste projeto de estágio era o desenvolvimento de um Sistema de Recomendação e de um Sistema de Relatórios e Análise de Tendências. Como o Playnify, um dos maiores projetos da empresa, está em fase de desenvolvimento contínuo, desenvolvemos apenas uma versão Demo para o Sistema de Recomendação, que será futuramente integrado com o Playnify. Para o Sistema de Relatórios e Análise de Tendências desenvolvemos um sistema novo que foi implementado com sucesso tanto pela parte de interface como das funcionalidades propostas.

As listas de recomendação geradas pelo Sistema de Recomendação correspondem ao objetivo inicialmente traçado. Testámos com sucesso as funcionalidades para cada categoria de recomendação (“Desportista”, “Eventos” e “Instalação desportiva”). O desempenho dos algoritmos implementados ainda não foi testado relativamente a tempos de execução. Como utilizámos SQL na implementação do algoritmo base, com poucos *joins* na recolha de dados da base de dados, acreditamos que tenha um desempenho satisfatório. A parte de avaliação do sistema também ainda não foi feita, devido ao problema de dispersão: mesmo quando se fizeram testes com utilizadores com o maior número de amigos (*top users*), alguns ainda ficaram sem recomendação para algumas categorias, não permitindo assim avaliar a qualidade/utilidade das recomendações.

Os relatórios do Sistema de Relatórios e Análise de Tendências desenvolvidos cumpriram todos os requisitos funcionais e não funcionais definidos anteriormente. O sistema permite-nos ainda a impressão dos relatórios através da página do sistema e a exportação para formatos externos, como PDF, XML, Microsoft Word e Microsoft Excel. Como desenvolvemos o sistema baseado na base de dados do Playnify (uma base de dados existente) não nos foi permitido alterar a arquitetura da base de dados para a

podermos utilizar de acordo com as nossas necessidades. Assim, encontrámos dificuldade em definir os filtros de relatório para “país”, “região” e “cidade” no Crystal Reports (CR), devido a estes registos estarem todos registados na mesma tabela da base de dados. Contudo, tomámos como solução uma outra alternativa onde definimos os filtros separados no ASP.NET e depois integramos com o CR, que permitiu o desenvolvimento com sucesso.

## 6.2. Trabalho Futuro

Relativamente ao Sistema de Recomendação seria desejável poder avaliar a qualidade/utilidade das recomendações geradas. Dois modelos de avaliação possíveis são:

- *Avaliação offline*: divide os dados atuais em passado e futuro, utiliza os dados passados para treino dos algoritmos e os dados do futuro para a avaliação. A comparação das recomendações a partir dos dados do passado com os dados do futuro, permite medir a qualidade dos algoritmos utilizados.
- *Avaliação online*: esta avaliação poderia ser feita quando o Sistema de Recomendação já estiver integrado com o Playnify no ambiente de produção. Com a avaliação *online* podemos medir até que ponto os utilizadores respondem positivamente em tempo real às recomendações geradas pelo sistema.

Está ainda previsto modificar os algoritmos de recomendação de forma a filtrarem os candidatos para recomendação (desportistas/eventos/instalações desportivas) pela distância à localização do utilizador ativo. Isto é, em vez de considerar todos os candidatos na mesma cidade, consideram-se os que se encontram dentro de um raio de distância predefinido.

Relativamente ao Sistema de Relatórios e Análise de Tendências identificámos os seguintes aspetos que não estiveram previstos no âmbito deste projeto de estágio:

- Um Sistema de Análise de Tendências que prevê as tendências futuras dos utilizadores com base nos dados históricos registados na base de dados.
- Uma funcionalidade que permite o envio automático de relatórios específicos aos responsáveis do sistema com base em horários de envio configurados.

## Referências

- [1] D. M. Boy e N. B. Ellison, "Social Network Sites: Definition, History and Scholarship," *Journal of Computer Mediated Communication*, vol. 13, n.º 1, 2007.
- [2] L. W. Liu, "TIME," 29 January 2008. [Online]. Available: <http://www.time.com/time/business/article/0,8599,1707760,00.html>. [Acedido em 19 Fevereiro 2013].
- [3] S. Steinberg, "Rollingstone," Maio 2011. [Online]. Available: <http://www.rollingstone.com/culture/blogs/gear-up/friendster-is-dead-encourages-u-s-users-to-move-on-20110511>. [Acedido em 19 Fevereiro 2013].
- [4] R. Matrins, "Estadão," 4 Setembro 2011. [Online]. Available: <http://blogs.estadao.com.br/rodrigo-martins/2011/09/04/facebook-ultrapassa-orkut-em-usuarios-e-acaba-com-reinado-de-7-anos-no-br/>. [Acedido em 19 Fevereiro 2013].
- [5] P. Sloan, "CNet," 22 Dezembro 2011. [Online]. Available: [http://news.cnet.com/8301-1023\\_3-57347187-93/pinterest-crazy-growth-lands-it-as-top-10-social-site/?tag=mncol;txt](http://news.cnet.com/8301-1023_3-57347187-93/pinterest-crazy-growth-lands-it-as-top-10-social-site/?tag=mncol;txt). [Acedido em 7 Fevereiro 2013].
- [6] P. Cvijikj e F. Michahelles, "Monitoring Trends on Facebook," em *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing*, Zurich, Switzerland, 2011.
- [7] M. Uchida e N. Shibata, "Identification and Visualization of Emerging Trends From Blogosphere," em *International Conference on Weblogs and Social Media*, Boulder, CO USA, 2006.
- [8] F. Fang e t. all, "Detecting Twitter Trends in Real-Time," em *Proceedings of the 21st Workshop on Information Technologies and Systems*, Johannesburg, South Africa, 2011.
- [9] J. Melià-Seguí e t. all, "Activity Duration Analysis for Context-aware Services Using Foursquare Check-ins," em *Proceedings of the 2012 International Workshop on Self-aware Internet of Things*, New York, 2012.
- [10] R. M. Bell e Y. Koren, "Scalable Collaborative Filtering with Jointly Derived Neighborhood," em *IEEE International Conference on Data Mining*, 2007.
- [11] G. Linden, B. Smith e J. York, "Amazon.com Recommendations: Item-to-Item

- Collaborative Filtering,” IEEE Computer Society, 2003.
- [12] “Visual Studio,” Microsoft, [Online]. Available: <http://www.microsoft.com/visualstudio/ptb/products/visual-studio-overview>. [Acedido em 19 Fevereiro 2013].
- [13] SAP, “SAP Crystal Reports 2011 Installation Guide,” SAP, 2010.
- [14] “HTML 5 Mozilla Developer Network,” Mozilla Developer Network, [Online]. Available: <https://developer.mozilla.org/en-US/docs/HTML>. [Acedido em 17 Fevereiro 2013].
- [15] “HTML5,” World Wide Web Consortium, [Online]. Available: <http://www.w3.org/TR/2012/CR-html5-20121217/>. [Acedido em 17 Fevereiro 2013].
- [16] “CSS Transitions,” World Wide Web Consortium, [Online]. Available: <http://www.w3.org/TR/2013/WD-css3-transitions-20130212/>. [Acedido em 17 Fevereiro 2013].
- [17] Jotorres, “Code Project,” [Online]. Available: <http://www.codeproject.com/Articles/534632/WhatplusisplusAJAX-3f>. [Acedido em 25 Fevereiro 2013].
- [18] “.Net Data Provider for PostgreSQL,” NPGSQL, [Online]. Available: <http://npgsql.projects.pgfoundry.org/>. [Acedido em 25 Fevereiro 2013].
- [19] I. Sommerville, “Requirements engineering,” em *Software Engineering*, edição 9 ed., Addison-Wesley, pp. 84-85, 2009.
- [20] I. Sommerville, “Extreme programming,” em *Software Engineering*, Edição 9 ed., Addison-Wesley, p. 65, 2009.
- [21] A. Cockburn, “Introduction to Use Cases,” em *Writing Effective Use Cases*, Addison-Wesley, pp. 15-16, 2001.
- [22] B. Sarwar, G. Karypis, J. Konstan e J. Riedl, “Item-Based Collaborative Filtering Recommendation Algorithms,” em *Proceedings of the 10th International Conference on World Wide Web*, Hong kong, 2001.



# Anexos

## A.1. O algoritmo de recomendação dos desportistas

```
public void players_bind()
{
    //Conectar a base de dados
    NpgsqlConnection conn = new NpgsqlConnection(connect());

    conn.Open();

    NpgsqlCommand cmd = new NpgsqlCommand("select FOAF.to_user_id as id_Players,
   USR.first_name||' '||USR.last_name as Sug_Players,
count(FOAF.to_user_id)/count(distinct(PSP.sport_id)) F_R_Players"

    + " from (select to_user_id" //Os amigos dos amigos do Active User***
        + " from friends_friendship"
        + " where from_user_id in(" // Os amigos do Active User
        (From_user_id) *
            + " select to_user_id"
            + " from friends_friendship"
            + " where from_user_id='" + username_bind() + "'"
            + " union"
            + " select from_user_id"
            + " from friends_friendship"
            + " where to_user_id='" + username_bind() + "'" //Fim do *
        + " union all"
        + " select from_user_id"
        + " from friends_friendship"
        + " where to_user_id in(" // Os amigos do Active User
        (To_user_id) **
            + " select to_user_id"
            + " from friends_friendship"
            + " where from_user_id='" + username_bind() + "'"
            + " union"
            + " select from_user_id"
            + " from friends_friendship"
            + " where to_user_id='" + username_bind() + "'" //Fim
        do * // Fim do ***
        + " inner join auth_user USR on FOAF.to_user_id = USR.id"
    //Juntar com Auth_user para buscar o nome
        + " inner join users_userprofile UUP on
    USR.id=UUP.user_id" //Juntar com users_userprofile para comunicar com
    Geo_localização
        + " inner join users_playerprofile UPP on
    UUP.user_id=UPP.userprofile_ptr_id" // Juntar para obter Availability
        + " inner join users_playersportprofile PSP on
    UPP.userprofile_ptr_id=PSP.user_id" //Juntar para obter o perfil do player
        + " where FOAF.to_user_id !='" + username_bind() + "'" and FOAF.to_user_id not
    in(" //Excluí os amigos do active user, e ele próprio ****
        + " select to_user_id"
        + " from friends_friendship"
        + " where from_user_id='" + username_bind() + "'"
        + " union"
        + " select from_user_id"
        + " from friends_friendship"
        + " where to_user_id='" + username_bind() + "'" // Fim do ****
```

```

+ " AND FOAF.to_user_id not in(" //Exclui os que já enviou os pedidos de
amizade *****
+ " select to_user_id"
+ " from friends_friendshipinvitation"
+ " where from_user_id='" + username_bind() + "'"
+ " union"
+ " select from_user_id"
+ " from friends_friendshipinvitation"
+ " where to_user_id='" + username_bind() + "'" // Fim do *****
+ " AND UUP.city_id = (select city_id" // Que perto do Active user (Na
mesma cidade)
+ " from users_userprofile
+ " where user_id='" + username_bind() + "'"
+ " AND PSP.sport_id in (select sport_id from users_playersportprofile
where user_id='" + username_bind() + "'" // Tem perfil do desporto que o Active
User pratica
+ " AND PSP.skill_level in (select skill_level from
users_playersportprofile where user_id='" + username_bind() + "'" //Tem a mesma
Skill Level do Active User
+ " AND (UPP.availability & (select availability from
users_playerprofile where userprofile_ptr_id='" + username_bind() + "'" > 0)" //
Tem a mesma disponibilidade com o Active User
+ " group by FOAF.to_user_id, Sug_Players"
+ " order by F_R_Players desc"
+ " limit 10", conn);

NpgsqlDataReader AmigoDr = cmd.ExecuteReader();
GridViewPlayers.DataSource = AmigoDr;
GridViewPlayers.DataBind();

conn.Close();
}

```

## A.2. User control da localização geográfica

Código do ASP.NET:

```

<%@ Control Language="C#" AutoEventWireup="true"
CodeBehind="LocationControl.ascx.cs" Inherits="Playnify_Reporting.LocationControl"
%>

<asp:ScriptManager ID="ScriptManagerLocal" runat="server">
</asp:ScriptManager>

<!-- Área de DropDownList do País-->
<asp:UpdatePanel ID="paispanel" runat="server">
<ContentTemplate>
<asp:Label ID="LblPais" runat="server" Text="País" Visible="False"
Width="41px"></asp:Label>
<asp:DropDownList ID="ddlPais" AutoPostBack = "true"
AppendDataBoundItems = "true" runat="server"
OnSelectedIndexChanged="ddlPais_SelectedIndexChanged"
Height="20px"
Width="160px" Visible="False">
</asp:DropDownList>
</ContentTemplate>
<Triggers>
<asp:AsyncPostBackTrigger ControlID="ddlPais"
EventName="SelectedIndexChanged" />
</Triggers>

```

```

</asp:UpdatePanel>
<!-- FIM Área de Dropdownlist do País-->

<!-- Área de Dropdownlist da Região -->
<asp:UpdatePanel ID="regiaopanel" runat="server">
    <ContentTemplate>
        <asp:Label ID="LblRegiao" runat="server" Text="Região"
Visible="False"></asp:Label>
        <asp:DropDownList ID="ddlRegiao" AutoPostBack="true"
AppendDataBoundItems="true"
runat="server"
OnSelectedIndexChanged="ddlRegiao_SelectedIndexChanged"
Height="20px" Width="160px" Visible="False">
        </asp:DropDownList>
    </ContentTemplate>
    <Triggers>
        <asp:AsyncPostBackTrigger ControlID="ddlRegiao"
EventName="SelectedIndexChanged" />
    </Triggers>
</asp:UpdatePanel>
<!-- FIM Área de Dropdownlist da Região -->

<!-- Área de Dropdownlist da Cidade -->
<asp:UpdatePanel ID="cidadepanel" runat="server">
    <ContentTemplate>
        <asp:Label ID="LblCidade" runat="server" Text="Cidade"
Visible="False"></asp:Label>
        <asp:DropDownList ID="ddlCidade" AutoPostBack="true"
AppendDataBoundItems="true"
runat="server" Height="20px" Width="160px" Visible="False">
        </asp:DropDownList>
    </ContentTemplate>
    <Triggers>
        <asp:AsyncPostBackTrigger ControlID="ddlCidade" />
    </Triggers>
</asp:UpdatePanel>
<!-- FIM Área de Dropdownlist da Região -->

```

#### Code Behind C#:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Npgsql;
using System.Data;

namespace Playnify_Reporting
{
    public partial class LocationControl : System.Web.UI.UserControl
    {
        // Conectar a base de dados
        NpgsqlConnection conn = new NpgsqlConnection("Server=
server.dev.ubiwhere.ian; Port=5432; Database=playnify_20130503; User
Id=user; Password=pass; SSL=false; SslMode=disable; encoding=Unicode");

        // Função para recuperar os países da base de dados
    }
}

```

```

public void Bind_C_ddlPais()
{
    conn.Open();

    NpgsqlCommand cmd = new NpgsqlCommand("select country, name from
geo_location where administration_level=0 order by name", conn);
    NpgsqlDataReader PReader = cmd.ExecuteReader();
    ddlPais.DataSource = PReader;
    ddlPais.Items.Clear();
    ddlPais.Items.Add("--- Selecione País ---");
    ddlPais.DataTextField = "name";
    ddlPais.DataValueField = "country";
    ddlPais.DataBind();

    conn.Close();
}

// Função para recuperar as regiões da base de dados
public void Bind_C_ddlRegiao()
{
    conn.Open();

    NpgsqlCommand cmd = new NpgsqlCommand("select id, name from
geo_location where administration_level=1 and country = '" + ddlPais.SelectedValue
+ "' order by name", conn);
    NpgsqlDataReader RReader = cmd.ExecuteReader();
    ddlRegiao.DataSource = RReader;
    ddlRegiao.Items.Clear();
    ddlRegiao.Items.Add("--- Selecione Região ---");
    ddlRegiao.DataTextField = "name";
    ddlRegiao.DataValueField = "id";
    ddlRegiao.DataBind();

    conn.Close();
}

//Função para recuperar as cidades da base de dados
public void Bind_ddlCidade()
{
    conn.Open();

    NpgsqlCommand cmd = new NpgsqlCommand("select id, name from
geo_location where administration_level=2 and CAST(parent_id as varchar) like '" +
ddlRegiao.SelectedValue + '%" order by name", conn);
    NpgsqlDataReader CReader = cmd.ExecuteReader();
    ddlCidade.DataSource = CReader;
    ddlCidade.Items.Clear();
    ddlCidade.Items.Add("--- Selecione Cidade ---");
    ddlCidade.DataTextField = "name";
    ddlCidade.DataValueField = "id";
    ddlCidade.DataBind();

    conn.Close();
}

protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        Bind_C_ddlPais();
    }
}

```

```
    }  
}  
  
protected void ddlPais_SelectedIndexChanged(object sender, EventArgs e)  
{  
    Bind_C_ddlRegiao();  
  
    ddlCidade.Items.Clear();  
}  
  
protected void ddlRegiao_SelectedIndexChanged(object sender, EventArgs e)  
{  
    Bind_ddlCidade();  
}  
  
//Função para parametro de filtro do DropDownList para País, Região e  
Cidade  
public void Parametro(int n)  
{  
    switch (n)  
    {  
        //País  
        case 1:  
            ddlPais.Visible = true;  
            lblPais.Visible = true;  
            break;  
  
        //Região  
        case 2:  
            ddlPais.Visible = true;  
            lblPais.Visible = true;  
            ddlRegiao.Visible = true;  
            lblRegiao.Visible = true;  
            break;  
  
        //Cidade  
        case 3:  
            ddlPais.Visible = true;  
            lblPais.Visible = true;  
            ddlRegiao.Visible = true;  
            lblRegiao.Visible = true;  
            ddlCidade.Visible = true;  
            lblCidade.Visible = true;  
            break;  
    }  
}  
  
// Função para retornar o id de nome do país selecionado do DropDownList  
public string pais()  
{  
    // Se não escolhe, retorna valor por omissão do país = "Portugal"  
    if (ddlPais.SelectedValue == "--- Selecione País ---")  
        return "PT";  
    else  
        return ddlPais.SelectedValue;  
}  
  
// Função para retornar o id de nome da região selecionado do DropDownList  
public string regiao()  
{  
    // Se não escolhe ou vazio, retorna valor por omissão da região =  
    "Porto"
```

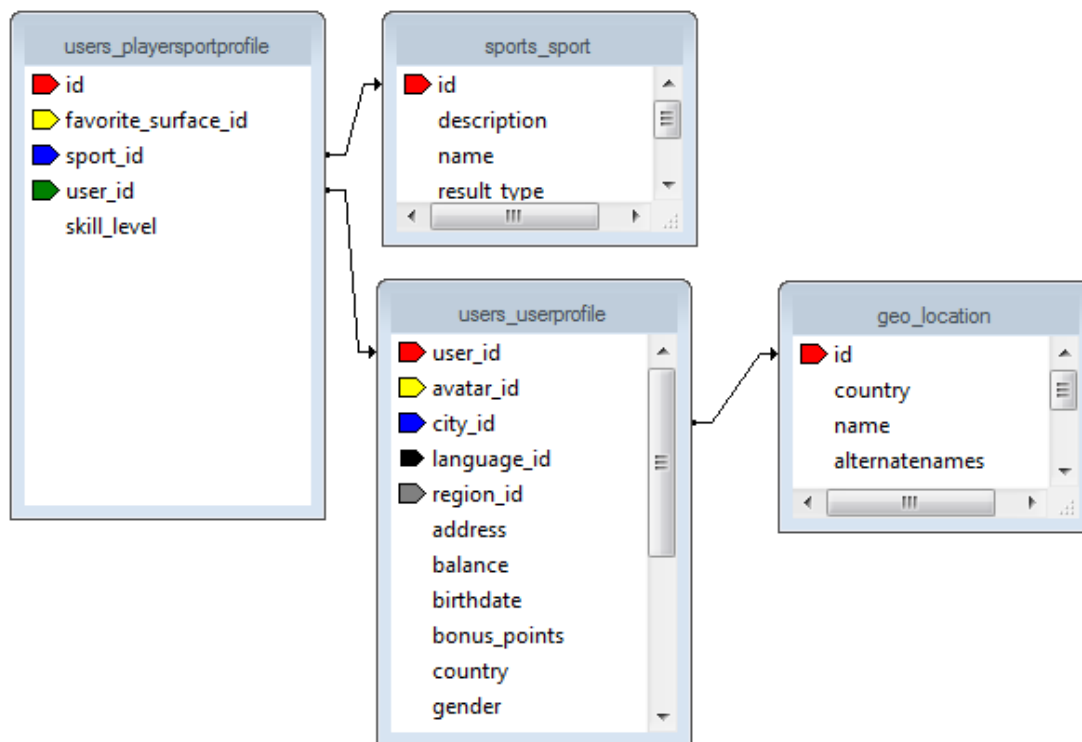
```

        if (ddlRegiao.SelectedValue == "" || ddlRegiao.SelectedValue == "---
Selezione Região ---")
            return "2735941";
        else
            return ddlRegiao.SelectedValue;
    }

    // Função para retornar o id de nome da região selecionado do DropDownList
    public string cidade()
    {
        // Se não escolhe ou vazio, retorna valor por omissão da cidade =
        "Porto"
        if (ddlCidade.SelectedValue == "" || ddlCidade.SelectedValue == "---
Selezione Cidade ---")
            return "6458924";
        else
            return ddlCidade.SelectedValue;
    }
}
}

```

### A.3. A Relação entre tabelas no CR



## A.4. Integração de Filtros e Relatórios na Página do Sistema

Código do ASP.NET:

```
<%@ Page Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
CodeBehind="Cidade_Frequencia_evento_desp_diadesemana.aspx.cs"
Inherits="Playnify_Reporting.Cidade_Frequencia_evento_desp_diadesemana" %>
<%@ Register Src="~/LocationControl.ascx" TagName="Cfedd" TagPrefix="C" %>
<%@ Register Src="~/DateControl.ascx" TagName="Dfedd" TagPrefix="D" %>
<%@ Register Src="~/SportsControl.ascx" TagName="Sfedd" TagPrefix="S" %>
<%@ Register assembly="CrystalDecisions.Web, Version=13.0.2000.0, Culture=neutral,
PublicKeyToken=692fbea5521e1304" namespace="CrystalDecisions.Web" tagprefix="CR"
%>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
    <div>
        <table class="tableStyle">
            <tr>
                <td><C:Cfedd ID="CFEDD" runat="server" /></td>
                <td><D:Dfedd ID="DFEDD" runat="server" /></td>
                <td><S:Sfedd ID="SFEDD" runat="server" /></td>
            </tr>
        </table>
    </div>

    <asp:Button ID="btnRelatorio" runat="server" Text="Gerar Relatório"
        onclick="btnRelatorio_Click" />
    <br />

    <CR:CrystalReportViewer ID="CidadeFrequenciaEventoDia" runat="server"
AutoDataBind="True"
        GroupTreeImagesFolderUrl="" HasCrystalLogo="False" HasDrilldownTabs="False"
        HasToggleGroupTreeButton="False" HasToggleParameterPanelButton="False"
        Height="962px" PrintMode="ActiveX" ReportSourceID="CrystalReportSource1"
        ToolbarImagesFolderUrl="" ToolPanelView="None" ToolPanelWidth="200px"
        Width="1344px" />
    <CR:CrystalReportSource ID="CrystalReportSource1" runat="server">
        <Report FileName="Cidade_Frequencia_evento_por_desp_diadesemana.rpt">
        </Report>
    </CR:CrystalReportSource>
</asp:Content>
```

Code behind C#:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using CrystalDecisions.CrystalReports.Engine;

namespace Playnify_Reporting
{
    public partial class Cidade_Frequencia_evento_desp_diadesemana :
        System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
```

```

{
    // Parametrizar para o nível de Cidade
    CFEDD.Parametro(3);
    //Esconder a janela do relatório
    CidadeFrequenciaEventoDia.Visible = false;
}

protected void btnRelatorio_Click(object sender, EventArgs e)
{
    // Visualizar a janela do relatório
    CidadeFrequenciaEventoDia.Visible = true;

    // Instanciar o relatório
    Cidade_Frequencia_evento_por_desp_diadesemana C_freq_eve_Desp_dia =
new Cidade_Frequencia_evento_por_desp_diadesemana();

    // Mandar o ID retornado do elemento selecionado da lista de filtro da
cidade ao parâmetro "Cidade" no relatório
Cidade_Frequencia_evento_por_desp_diadesemana.rpt
C_freq_eve_Desp_dia.SetParameterValue("Cidade", CFEDD.cidade());

    // Mandar a data retornado da data início selecionada no filtro de
data ao parâmetro "Dt_ini" no relatório
Cidade_Frequencia_evento_por_desp_diadesemana.rpt
C_freq_eve_Desp_dia.SetParameterValue("Dt_ini", DFEDD.data_inicio());

    // Mandar a data retornado da data fim selecionada no filtro de data
ao parâmetro "Dt_fim" no relatório
Cidade_Frequencia_evento_por_desp_diadesemana.rpt
C_freq_eve_Desp_dia.SetParameterValue("Dt_fim", DFEDD.data_fim());

    // o ID retornado do elemento selecionado da lista de filtro do
desporto ao parâmetro "Sports" no relatório
Cidade_Frequencia_evento_por_desp_diadesemana.rpt
C_freq_eve_Desp_dia.SetParameterValue("Sports", SFEDD.sports());

    //Definir o relatório
Cidade_Frequencia_evento_por_desp_diadesemana.rpt como fonte de relatório
CidadeFrequenciaEventoDia.ReportSource = C_freq_eve_Desp_dia;
}
}
}

```

Sintaxe do CR na Fórmula de seleção do relatório para interpretar o filtro

```

// Transforma o valor recebido do filtro de localização em número inteiro
// porque o ID retornado é um string
{geo_location.id} = ToNumber({?Cidade}) and

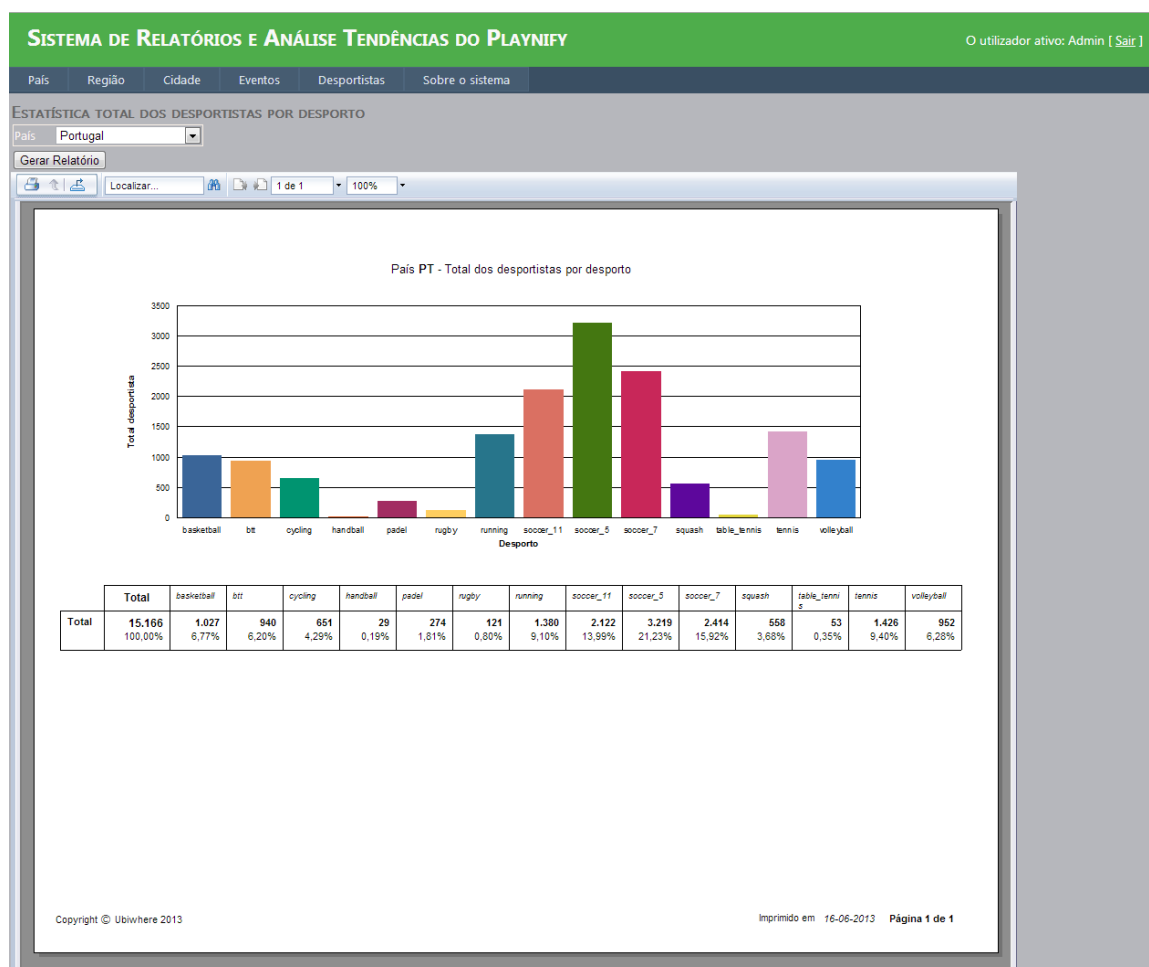
// Transforma o valor recebido do filtro de desporto em número inteiro
// porque o ID retornado é um string - se o valor é 'todos' então não aplica o filtro
(if {?Sports} <> "todos" then {sports_sport.id}=ToNumber({?Sports}) else true) and

// Transforma o valor recebido do filtro de datas em data e hora
// porque o ID retornado é um string
{events_eventmatch.end_date} >= datetime({?Dt_ini}) and
{events_eventmatch.end_date}<=datetime({?Dt_fim})

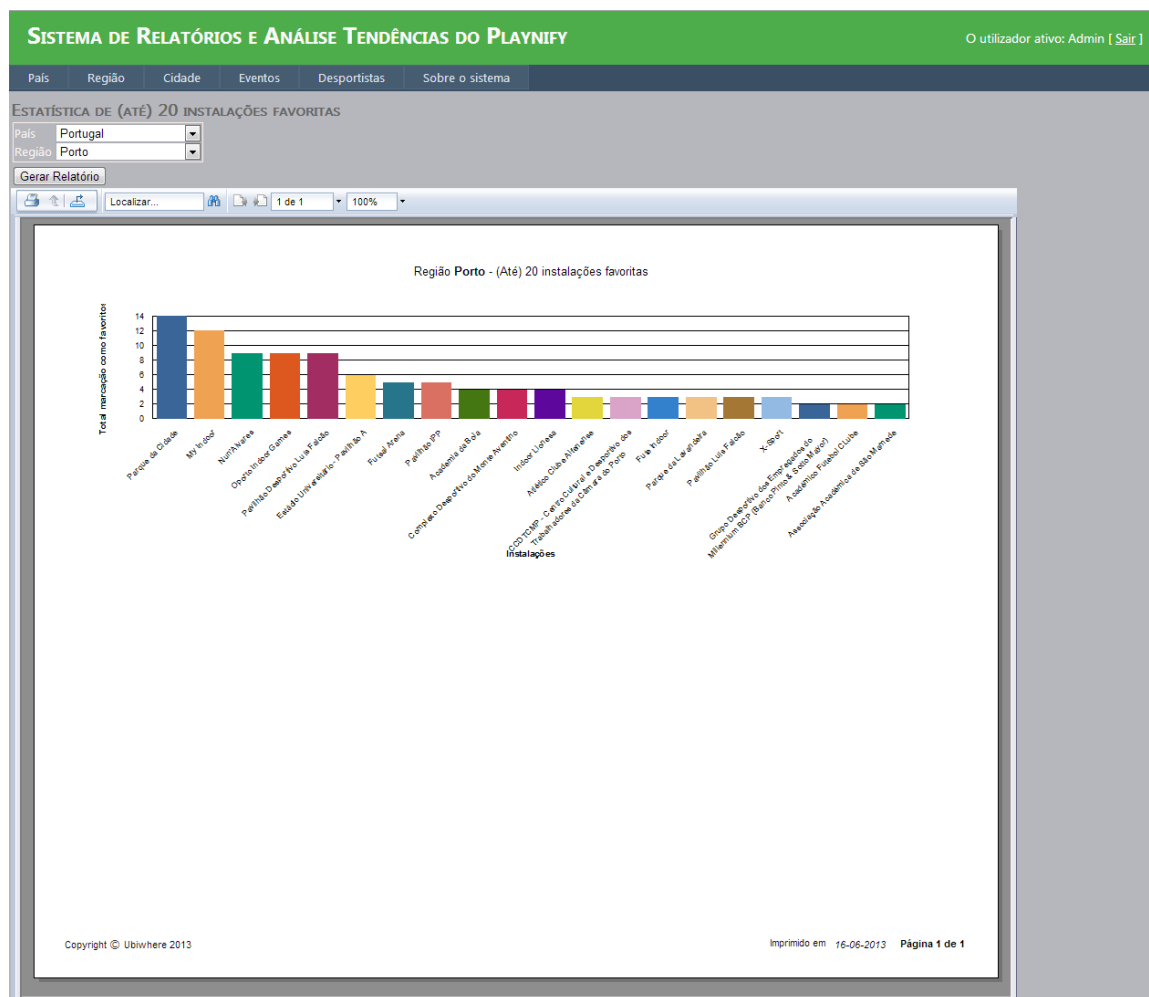
```



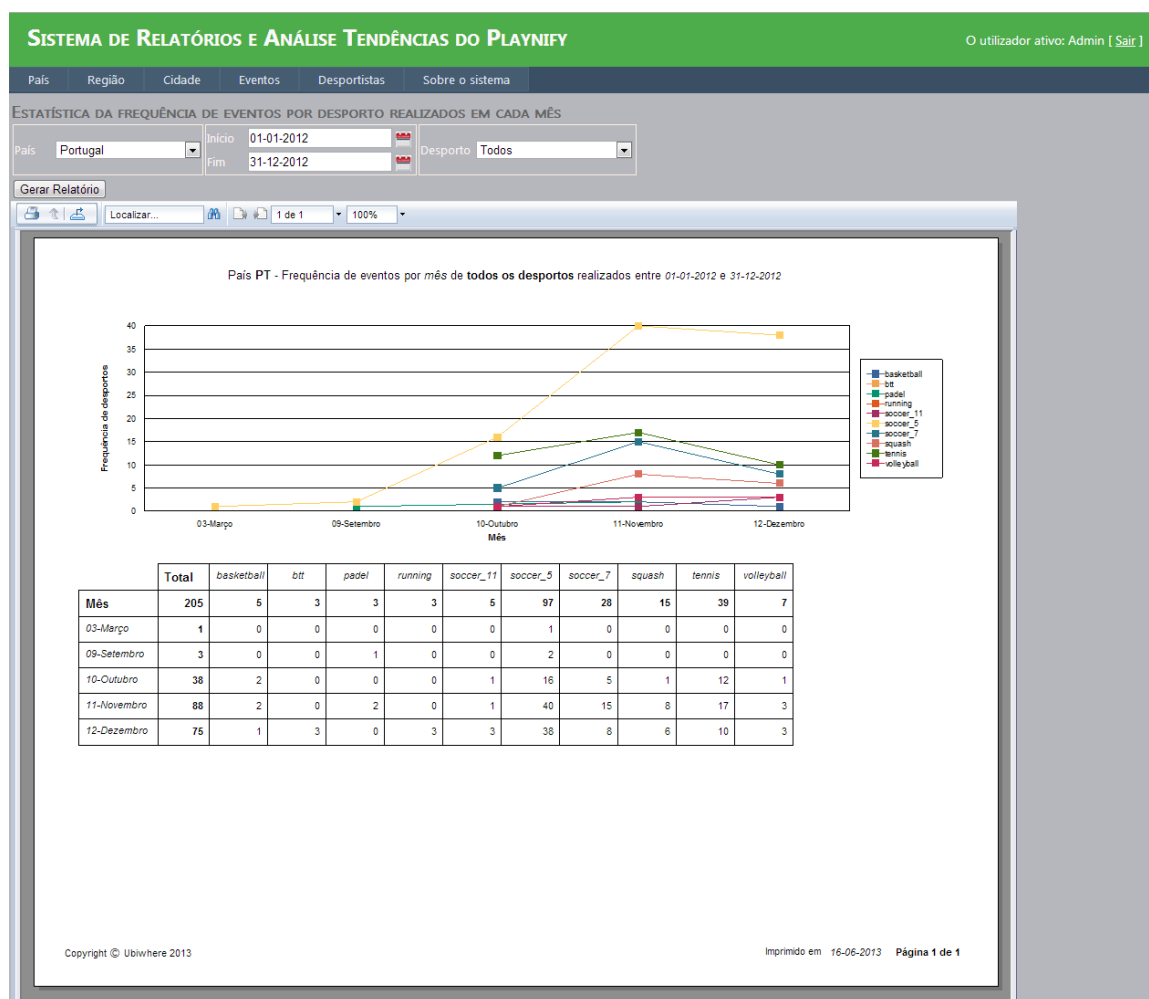
## A.5. Uma Estatística de Relatório dos Desportistas por Desporto



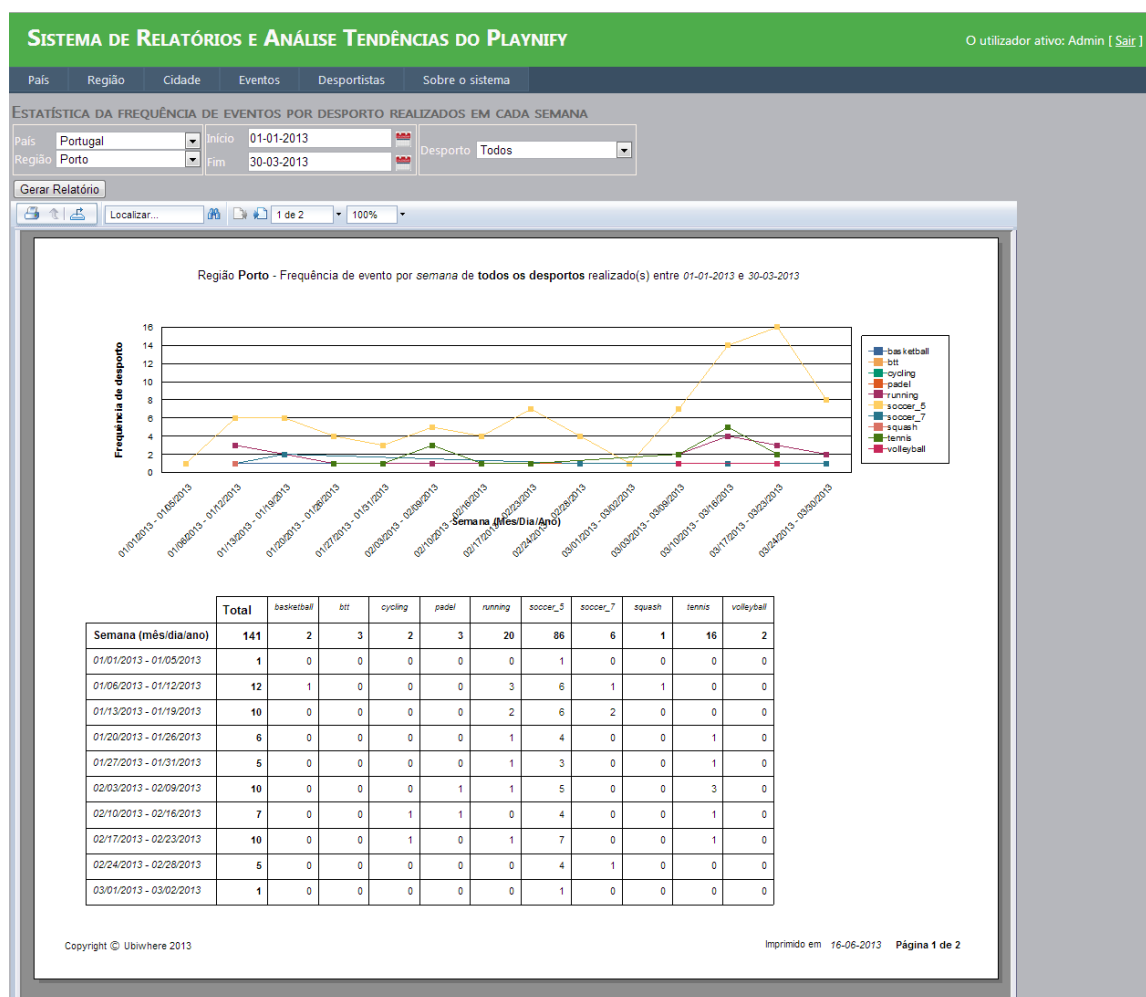
## A.6. Uma Estatística de Relatório das Instalações Desportivas



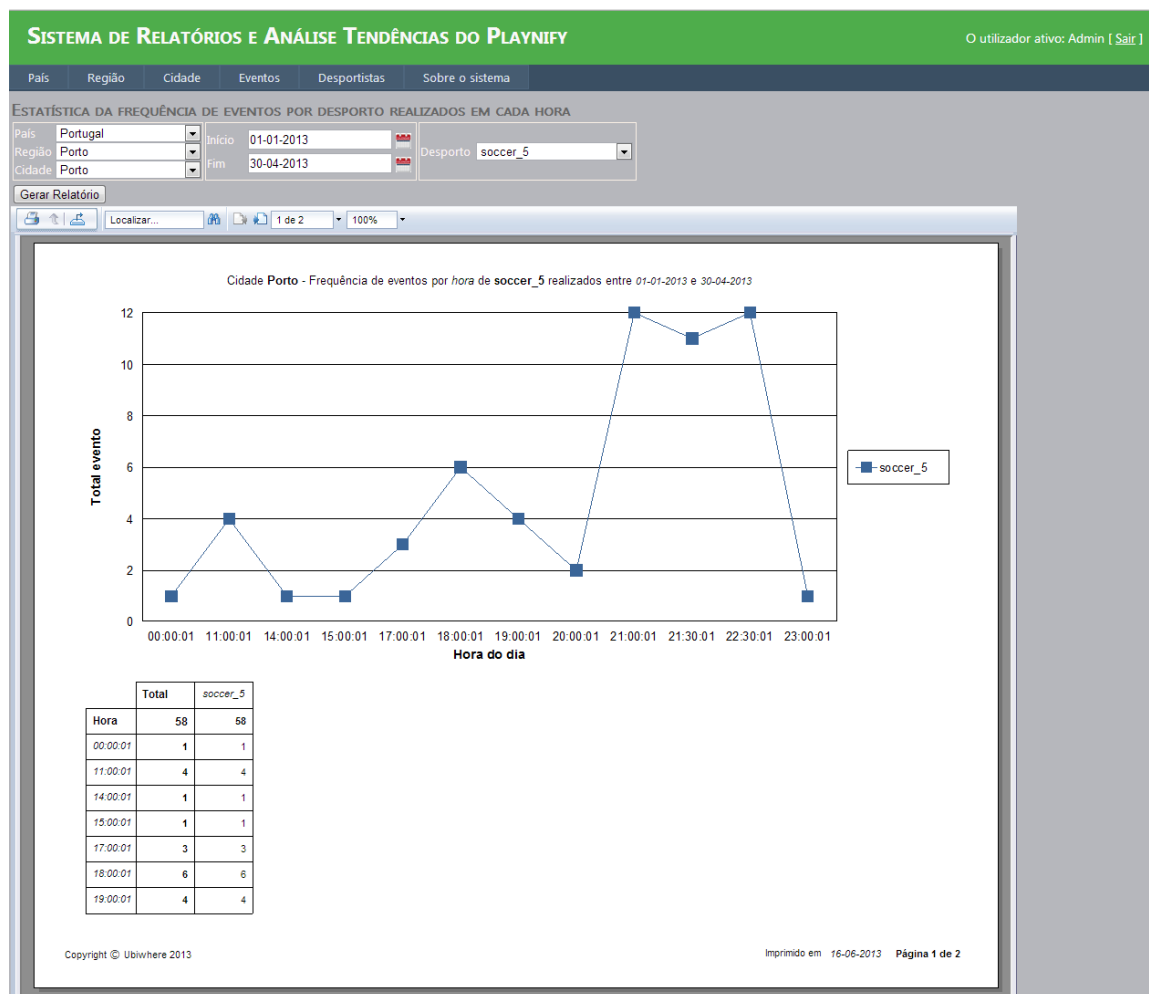
## A.7. Uma Estatística de Análise de Tendências Categorizada por Mês



## A.8. Uma Estatística de Análise de Tendências categorizada por Semana

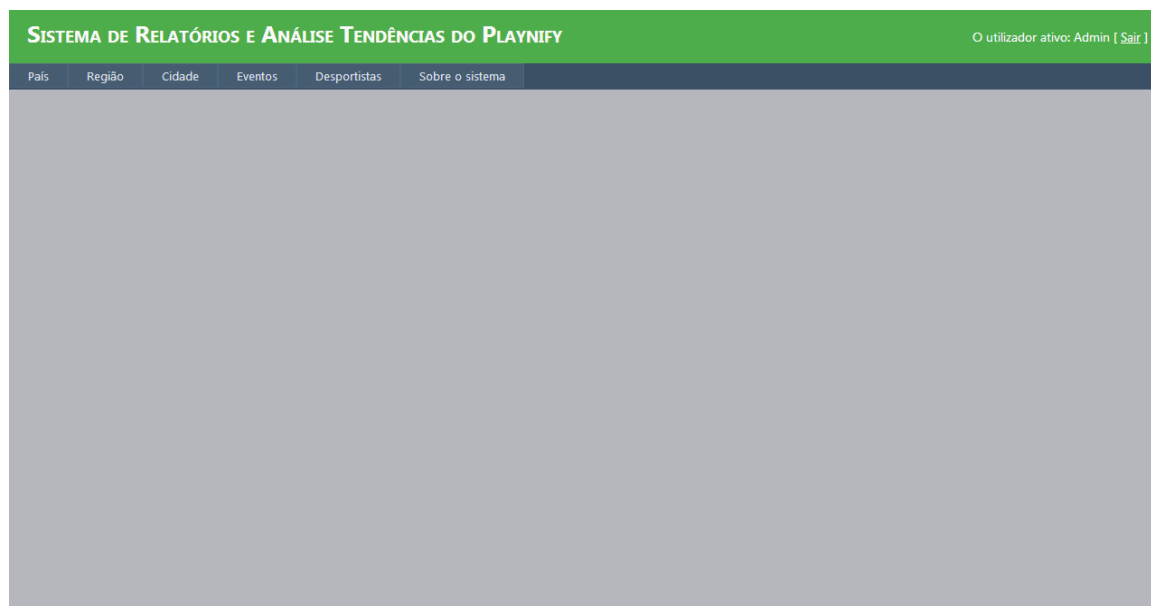


## A.9. Uma Estatística de Análise de Tendências categorizada por Hora do Soccer\_5 (Futsal)



## A.10. Página principal, Menus e Submenus do Sistema de Relatórios e Análise de Tendências

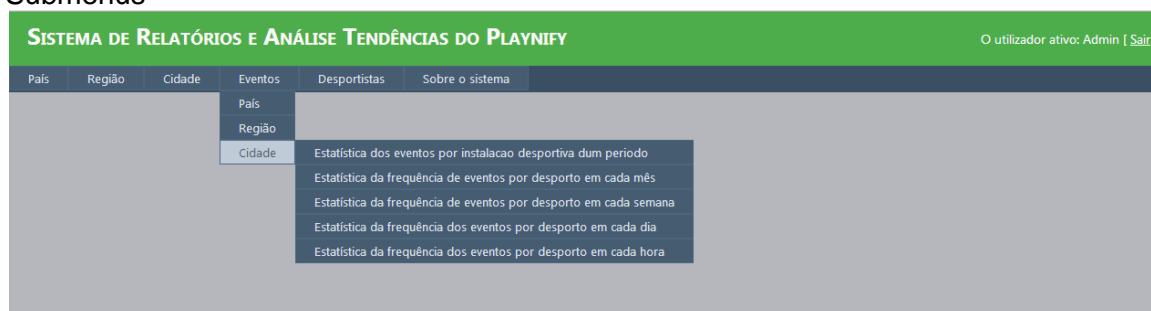
### Página principal



### Menus



### Submenus



## A.11. Ficheiro de Configuração do Sistema de Relatórios e Análise de Tendências

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>

  <appSettings>
    <add key="CrystalImageCleaner-AutoStart" value="true" />
    <add key="CrystalImageCleaner-Sleep" value="60000" />
    <add key="CrystalImageCleaner-Age" value="120000" />
  </appSettings>

  <!-- Conexão com ODBC driver -->
  <connectionStrings>
    <add name="PgConnect" connectionString="Dsn=PostgreSQL30;Uid=user;Pwd=pass"
providerName="System.Data.Odbc" />
  </connectionStrings>

  <!-- Configuração Assemblies do Crystal Reports -->
  <system.web>
    <compilation debug="true" optimizeCompilations="true" defaultLanguage="c#"
targetFramework="4.0">
      <assemblies>
        <add assembly="CrystalDecisions.Web, Version=13.0.2000.0, Culture=neutral,
PublicKeyToken=692FBEA5521E1304" />
        <add assembly="CrystalDecisions.Shared, Version=13.0.2000.0,
Culture=neutral, PublicKeyToken=692FBEA5521E1304" />
        <add assembly="log4net, Version=1.2.10.0, Culture=neutral,
PublicKeyToken=692FBEA5521E1304" />
        <add assembly="CrystalDecisions.ReportSource, Version=13.0.2000.0,
Culture=neutral, PublicKeyToken=692FBEA5521E1304" />
        <add assembly="CrystalDecisions.ReportAppServer.Controllers,
Version=13.0.2000.0, Culture=neutral, PublicKeyToken=692FBEA5521E1304" />
        <add assembly="CrystalDecisions.ReportAppServer.DataDefModel,
Version=13.0.2000.0, Culture=neutral, PublicKeyToken=692FBEA5521E1304" />
        <add assembly="CrystalDecisions.CrystalReports.Engine,
Version=13.0.2000.0, Culture=neutral, PublicKeyToken=692FBEA5521E1304" />
        <add assembly="CrystalDecisions.ReportAppServer.ClientDoc,
Version=13.0.2000.0, Culture=neutral, PublicKeyToken=692fbea5521e1304" />
      </assemblies>
    </compilation>

    <httpHandlers>
      <add verb="GET" path="CrystalImageHandler.aspx"
type="CrystalDecisions.Web.CrystalImageHandler, CrystalDecisions.Web,
Version=13.0.2000.0, Culture=neutral, PublicKeyToken=692fbea5521e1304" />
    </httpHandlers>

    <customErrors mode="Off" defaultRedirect="GenericErrorPage.htm">
      <error statusCode="403" redirect="NoAccess.htm" />
      <error statusCode="404" redirect="FileNotFound.htm" />
    </customErrors>

    <authentication mode="Forms">
      <forms cookieless="UseCookies" defaultUrl="~/Account/Login.aspx"
loginUrl="~/Account/Login.aspx" protection="All" timeout="2880" />
    </authentication>
  </system.web>

  <system.webServer>
    <defaultDocument>
      <files>
        <clear />
      </files>
    </defaultDocument>
  </system.webServer>
</configuration>
```

```

        <add value="~/Account/Login.aspx" />
    </files>
</defaultDocument>

<modules runAllManagedModulesForAllRequests="true" />
<handlers>
    <add name="CrystalImageHandler.aspx_GET" verb="GET"
path="CrystalImageHandler.aspx" type="CrystalDecisions.Web.CrystalImageHandler,
CrystalDecisions.Web, Version=13.0.2000.0, Culture=neutral,
PublicKeyToken=692fba5521e1304" preCondition="integratedMode" />
</handlers>
<validation validateIntegratedModeConfiguration="false" />
</system.webServer>
</configuration>

```

## A.12. Os Queries da validação de Recomendação dos “Desportistas”

```

-- Obter o ID do utilizador ativo
-----
Select id
from auth_user
where email='ricrdoferrolho@gmail.com'

-- Confirmar a localização geográfica - Cidade [Resultado esperado = 1]
-----
select count(*)
from(
    Select city_id
    from users_userprofile
    where user_id='2022'
    intersect
    Select city_id
    from users_userprofile
    where user_id='5466'
) as Test_Location

-- Confirmar o perfil e o nível [Resultado esperado > 0]
-----
select count(*)
from(
    Select sport_id, skill_level
    from users_playersportprofile
    where user_id='2022'
    intersect
    Select sport_id, skill_level
    from users_playersportprofile
    where user_id='5466'
) as P_ProfSkill

-- Confirmar a disponibilidade [Resultado esperado = 1]
-----
select count(*)
from(
    select userprofile_ptr_id
    from users_playerprofile
    where availability & (select availability from users_playerprofile where
userprofile_ptr_id='2022') > 0
    intersect

```



```
        select userprofile_ptr_id
        from users_playerprofile
        where userprofile_ptr_id='5466'
) as P_Disponibilidade

--Confirmar número de amigos em Comum [Resultado esperado = O número de amigo
que se mostra no Friends Rate do Protótipo]
-----
select count (*)
from(
    select *
    from(
        select to_user_id
        from friends_friendship
        where from_user_id='2022'
        union
        select from_user_id
        from friends_friendship
        where to_user_id='2022') as F_ActUsr

    intersect
    select *
    from(
        select to_user_id
        from friends_friendship
        where from_user_id='5466'
        union
        select from_user_id
        from friends_friendship
        where to_user_id='5466') as F_RecUsr
) as F_CommonNum
```

### A.13. Os Queries da Validação de Relatório "Total desportistas por desporto em Portugal"

```
select S.name, count(*)
from users_userprofile UUP
    inner join users_playersportprofile UPSP on UPSP.user_id=UUP.user_id
    inner join sports_sport S on UPSP.sport_id=S.id
where UUP.country='PT'
group by S.name
order by S.name
```